WHICH MODELS TO MIX DEPENDS ON HOW YOU MIX THEM: COMPOSITION-AWARE PORTFOLIO SELECTION FOR MIXTURE-OF-MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Given a large pool of candidate language models, which small subset should be composed into a Mixture-of-Models? Existing rules score candidates by individual accuracy, error diversity, or complementary coverage, and implicitly value a pool by the probability that at least one member is correct. This assumption fails for an entire class of deployed systems. We distinguish *selection*-type composers, which output a member's answer, from *assembly*-type composers, which synthesize an answer from member evidence, and show that the two induce structurally different portfolio objectives. Under a subgoal decomposition of difficulty, the gap between their ceilings equals the probability that the members solve a query collectively while none solves it individually. This yields a dichotomy: the ceilings coincide exactly when this event is impossible; otherwise, the optimal portfolios can be disjoint, and using either in place of the other incurs a constant loss. Tractability is similarly asymmetric: the selection objective is monotone submodular at every information level, whereas the assembly objective exhibits increasing returns and admits no polynomial-time constant-factor approximation unless P = NP. Therefore, we recover a guarantee through a submodular logarithmic surrogate. Graders observe competence only after a member's upstream errors have propagated, so we build the surrogate on intrinsic competence, recovered in closed form by inversion over the prerequisite order. Without it, a selector can discard specialists that make assembly worthwhile. Our method performs best on six benchmarks with a 42-model pool, and its diagnostic indicates whether a homogeneous or heterogeneous portfolio performs better before composition runs.

1 INTRODUCTION

The number of usable language models now exceeds what any single deployment can accommodate: public routing corpora index tens of models (Hu et al., 2024; Li et al., 2026a), and one benchmark aggregates records for more than eight thousand (Huang et al., 2025). Deployments that serve queries with multiple models must therefore answer a question that precedes routing: which few candidates should be retained, kept warm, contracted for, and integrated into a composition? We call this problem *model portfolio selection*, after the algorithm portfolio that posed the same question for solvers (Rice, 1976; Xu et al., 2008). The choice matters in practice because performance need not improve monotonically with pool size. A unified routing benchmark reports clear diminishing returns from adding models and finds that a carefully chosen subset can outperform the full pool (Li et al., 2026a). Existing selection rules fall into three families: selecting the strongest candidates (Shnitzer et al., 2023); selecting candidates with decorrelated errors, following the ensemble diversity tradition (Kuncheva & Whitaker, 2003); and selecting candidates that rescue an anchor's failures, with a greedy guarantee from the submodularity of complementary coverage (Shihab et al., 2026).

All three rules center on the probability that at least one member answers correctly, and recent work has characterized this quantity precisely. Any policy whose output equals a member's answer is capped at one minus the joint failure rate, and average pairwise error correlation provably cannot identify this rate (Chen, 2026). The cap substantially exceeds the accuracy of the best fixed model, yet deployable routers capture only a fraction of the gap (Shihab et al., 2026). They plateau far below the cap because they lack instance-level correctness prediction (Lu et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Two implications follow. The first is widely recognized: coverage overstates what a router can deliver, because the router must identify a correct member before seeing any answers. The second, our starting point, is that for a growing class of systems coverage is not even an upper bound on accuracy. Mixture-of-Agents feeds member responses to a summarizer that generates a new answer (Wang et al., 2025), and related forms of synthesis appear in generative fusion (Jiang et al., 2023), debate (Du et al., 2024), and layered architectures found by inference-time search (Saad-Falcon et al., 2024). Such composers can answer correctly even when every member is wrong, so their accuracy can exceed coverage. A counterexample in Zhang et al. (2026) makes this concrete: two proposers, each uninformative about the truth on its own and highly correlated with the other, become jointly sufficient when combined with the summarizer's own signal. This explains why that work finds that the most complementary proposer can be individually weak.

Portfolio selection is therefore underdetermined until the composition operator is specified. There is no composer-agnostic notion of the best portfolio, because the two natural operator classes induce objectives that differ in mathematical structure, not merely in their parameters. This distinction also resolves a standing contradiction. Mixture-of-Agents reports that mixing model families outperforms any single model (Wang et al., 2025), whereas Self-MoA finds under controlled conditions that a homogeneous team of the best model consistently wins (Li et al., 2026b). Our dichotomy accommodates both findings as two regimes separated by a quantity that neither paper reports.

We summarize our contributions as follows:

- **A dichotomy between composer classes.** We show that *selection*- and *assembly*-type composers induce distinct value functionals whose ceilings differ by exactly the probability of distributed solvability. Their optimal portfolios can then be disjoint with constant loss both ways, which neither rules on the marginal law of correctness nor pool-level thresholds can detect.
- **A tractability asymmetry.** We prove selection is monotone submodular across information levels, while assembly has increasing returns and resists constant-factor approximation unless P = NP. We recover a guarantee through a submodular logarithmic surrogate with a computable certificate.
- **A propagation correction.** We recover intrinsic competence from graded competence in closed form, yielding a uniformly tighter surrogate that preserves submodularity. Without it, the bias grows quadratically with prerequisite depth, causing the specialists worth retaining discarded.
- **CAPS, and an evaluation protocol.** We propose CAPS, which scales through low-rank subgoal completion, debiases the plug-in maximum with a cross-fitted split-argmax estimator, and chooses the branch by comparing the optimized portfolios on calibration data. We report the ladder of ceilings alongside realized value, rather than a single accuracy metric.

2 RELATED WORK

Multi-Model Routing and Composition. Routing selects a member via confidence-based cascades (Chen et al., 2024), preference-trained strong–weak routers (Ong et al., 2025), or prompt-based multi-way utility prediction (Shnitzer et al., 2023; Zhuang et al., 2025; Song et al., 2025). Benchmarks standardize evaluation on precomputed outcomes (Hu et al., 2024; Huang et al., 2025; Li et al., 2026a), while classifier failures constrain the tested hypothesis class rather than achievable gain (Srivatsa et al., 2024). A second line synthesizes answers through candidate ranking and fusion (Jiang et al., 2023), critique-based debate (Du et al., 2024), and Mixture-of-Agents with layered proposers and a summarizer (Wang et al., 2025); Self-MoA questions the need for heterogeneous members (Li et al., 2026b). Inference-time and agentic search optimize techniques, prompts, and topologies (Saad-Falcon et al., 2024; Hu et al., 2025; Zhang et al., 2025). Sparse Mixture-of-Experts routes within a jointly trained network (Shazeer et al., 2017), whereas our candidates are frozen, independently trained, and separately priced. These methods take the pool as given; operator search treats model assignment as a categorical variable, overlooking how best assignment is determined.

Participant Selection. Closest to us, Zhang et al. (2026) study combinatorial proposer complementarity, show that adding proposers can hurt summarizer accuracy, and propose greedy methods based on marginal contributions. Turkmen et al. (2026) greedily maximize mutual information under a Gaussian copula model of correlated errors and characterize a saturation floor. Shihab et al. (2026) establish monotone submodularity of complementary coverage relative to a fixed anchor, yet find that greedy coverage fails to yield router gains. These objectives measure outcome coverage

or label-level information, and both lie on the selection ladder. We explain why coverage fails to translate into routing gains, and we give assembly its own objective, hardness result, and algorithm.

3 PRELIMINARIES

Pool, anchor, and subgoals. Let $\mathcal{M} = \{1, \dots, K\}$ be a frozen candidate pool and $X \sim \mathcal{D}$ a query with answer space $\mathcal{A}$ and truth Y. For a portfolio $S \subseteq \mathcal{M}$ and a composition operator Φ, we write $S_\Phi = S \cup \{\Phi\}$ and treat Φ as the deployment anchor, following Shihab et al. (2026). For assembly, Φ is itself a model with its own competence. Each query has $L = L(X)$ latent subgoals, member m has binary competence $B_{m\ell} \in \{0, 1\}$ on subgoal ℓ, and correctness is conjunctive:

$$C_m \;=\; \prod_{\ell=1}^{L} B_{m\ell} \;\in\; \{0, 1\}. \tag{1}$$

Eq. (1) is our only structural assumption. Subgoals are the units at which credit transfers between models: code tests, whose conjunction defines passing (Liu et al., 2023; Jain et al., 2025); annotated multi-hop sub-questions (Trivedi et al., 2022; Yang et al., 2018); or graded mathematical proof steps (Lightman et al., 2024; Hendrycks et al., 2021). Atomic multiple-choice items have $L = 1$ and recover the standard binary setting.

Error propagation. Let $G = ([L], E)$ be the per-query prerequisite directed acyclic graph (DAG), with subgoals in topological order and closed ancestor sets $\mathrm{an}^*(\ell) = \mathrm{an}(\ell) \cup \{\ell\}$. Member m's *intrinsic* competence $A_{m\ell} \in \{0, 1\}$ indicates whether it can solve subgoal ℓ given correct outputs on $\mathrm{an}(\ell)$. Its graded, or *propagated*, competence also reflects its own upstream failures:

$$B_{m\ell} \;=\; A_{m\ell} \prod_{j \in \mathrm{an}(\ell)} B_{mj} \;=\; \prod_{j \in \mathrm{an}^*(\ell)} A_{mj}. \tag{2}$$

Induction in topological order, together with binarity, gives the second equality and $C_m = \prod_\ell B_{m\ell} = \prod_\ell A_{m\ell}$. Eq. (2) thus refines Eq. (1), and $B = A$ when the DAG has no edges. Graders observe B, so an upstream failure can mask competence that becomes usable once the prefix is repaired: a model may fail a prerequisite yet solve a dependent subgoal when the composer supplies the correct prerequisite output. The objective needs A, but the data provide B.

Composer classes and value functionals. A composer maps $(X, \{R_m\}_{m \in S_\Phi})$ to an answer, where $R_m = (\hat{y}_m, E_m)$ is member m's answer together with its evidence. We call Φ *selection type* if its output always lies in $\{\hat{y}_m : m \in S_\Phi\}$, and *assembly type* if it can range over all of $\mathcal{A}$. Routers, cascades, voting, judge-based picking, and rerankers are of selection type; summarizer-based Mixture-of-Agents, generative fusion, and debate followed by synthesis are of assembly type. Let $\mathcal{G}$ and $\mathcal{H}$ be the sigma-fields that the two classes may condition on. With $\mu_m(\mathcal{G}) = \mathbb{E}[C_m \mid \mathcal{G}]$ and $\nu_{m\ell}(\mathcal{H}) = \mathbb{P}[B_{m\ell} = 1 \mid \mathcal{H}]$, the best achievable accuracy in each class is

$$F_{\mathcal{G}}(S) = \mathbb{E}\Big[\max_{m \in S_\Phi} \mu_m(\mathcal{G})\Big], \tag{3}$$

$$V_{\mathcal{H}}(S) = \mathbb{E}\Big[\max_{\pi:[L]\to S_\Phi} \mathbb{E}\big[\textstyle\prod_\ell B_{\pi(\ell)\ell} \,\big|\, \mathcal{H}\big]\Big], \tag{4}$$

where Eq. (4) maximizes over $\mathcal{H}$-measurable assignments of subgoals to members. Three information levels matter. A pre-answer signal Z gives the deployable selection value F_Z. Full outcome information gives the *coverage ceiling* $C(S) = \mathbb{P}[\exists m \in S_\Phi : C_m = 1]$. Full subgoal information gives two assembly ceilings, depending on whether the composer can only adopt sub-solutions as produced or can also re-feed a substituted one downstream:

$$V^\dagger(S) = \mathbb{E}\Big[\prod_{\ell=1}^{L} \max_{m \in S_\Phi} B_{m\ell}\Big], \qquad V^\ddagger(S) = \mathbb{E}\Big[\prod_{\ell=1}^{L} \max_{m \in S_\Phi} A_{m\ell}\Big]. \tag{5}$$

Both the *splice-free* and the *splice-capable* ceiling are realized by concrete systems. Concatenating candidate answers under a summarizer is splice-free, whereas a pipeline that exposes intermediate results and re-executes downstream steps on a repaired prefix is splice-capable. Replacing B with A in Eq. (4) defines the splice-capable value $V^\ddagger_{\mathcal{H}}$, whose full-information limit is $V^\ddagger$. Conditional Jensen on the maximum gives the ladder $\max_{m \in S_\Phi} \mathbb{E}C_m \le F_Z(S) \le C(S)$ and $V_{\mathcal{H}}(S) \le V^\dagger(S) \le V^\ddagger(S)$ (Lemma 12, Appendix A). Coverage thus sits at the top of the selection ladder, and coverage-driven selection optimizes an upper bound rather than the deployed quantity. The rest of the paper shows that coverage and Eq. (5) are different set functions.

4 THE TWO BRANCHES ARE DIFFERENT SET FUNCTIONS

4.1 SELECTION IS SUBMODULAR AT EVERY INFORMATION LEVEL

Theorem 1 (Uniform submodularity)**.** *For any sigma-field $\mathcal{G}$, the map $S \mapsto F_{\mathcal{G}}(S) - F_{\mathcal{G}}(\emptyset)$ is normalized, monotone, and submodular. Greedy under a cardinality budget k attains a $(1 - 1/e)$ fraction of the optimum, and cost-benefit greedy attains the same factor under a knapsack budget.*

This contains the coverage result of Shihab et al. (2026) as the case of binary conditional probabilities. Stating it at general $\mathcal{G}$ matters because the *deployable* objective F_Z inherits the guarantee. There is thus no reason to optimize the ceiling as a proxy, which closes a gap named in limitations of that work. The knapsack extension is classical (Sviridenko, 2004) and we claim no novelty for it.

4.2 ASSEMBLY: AN EXACT IDENTITY AND A DICHOTOMY

The assembly ceiling can exceed coverage when individually incorrect members provide complementary sub-solutions. The following identity quantifies this gap.

Proposition 2 (Distributed-solvability identity)**.** *For every $S \subseteq \mathcal{M}$, writing $B_m = (B_{m1}, \ldots, B_{mL})$ and taking $\vee$ componentwise,*

$$V^{\dagger}(S) - C(S) \;=\; \mathbb{P}\Big[\bigvee_{m \in S_\Phi} B_m = \mathbf{1} \;\wedge\; \forall m \in S_\Phi : B_m \neq \mathbf{1}\Big] \;=:\; \Delta(S) \;\geq\; 0. \tag{6}$$

We call $\Delta(S)$ the decomposability gap.

As with the complementary-correctness identity of Shihab et al. (2026), the value of Eq. (6) is that it names the event to be estimated. Coverage undercounts assembly by exactly the mass of queries that the portfolio solves collectively but not individually. No function of accuracy, diversity, or coverage sees that mass, because every member is wrong on those queries.

Theorem 3 (Ceiling dichotomy)**.** *$V^{\dagger}(S) = C(S)$ for all $S \subseteq \mathcal{M}$ if and only if, almost surely and for every S, collective solvability implies individual solvability, that is $\bigvee_{m \in S_\Phi} B_m = \mathbf{1}$ implies $\exists m \in S_\Phi : B_m = \mathbf{1}$. This condition is implied by, but strictly weaker than, the competence vectors of S_Φ forming a chain under the componentwise order, and it holds whenever $L = 1$.*

This is the formal content of the title. When collective solvability implies individual solvability, the two classes share a ceiling and, at the level of ceilings, the portfolio problem does not depend on the composer. Otherwise the ceilings differ, and the divergence is not a technicality.

Theorem 4 (Separation of optimal portfolios)**.** *There is an instance with $K = 4$, $L = 2$, $k = 2$ and a pre-answer signal Z on which $\arg\max_{|S|=2} F_Z$ and $\arg\max_{|S|=2} V^{\dagger}$ are disjoint, with F_Z equal to 0.8 at the selection optimum and 0 at the assembly optimum, and $V^{\dagger}$ equal to 1 at the assembly optimum and 0.8 at the selection optimum. Deploying the selection-optimal portfolio under an assembly composer therefore loses 0.2, and the reverse loses 0.8.*

The construction pairs two *identifiable generalists* with two *disjoint specialists*. The generalists are perfectly correlated, solve both subgoals with probability 0.8, and are predictably correct given Z; the specialists have deterministic competence $(1, 0)$ and $(0, 1)$. The specialists have zero individual accuracy, so every existing rule discards them, yet they are jointly sufficient, so an assembly composer should keep exactly them. Unlike a parity construction, this pattern is routinely observed: one model sets up a problem reliably but fumbles the arithmetic, while another does the reverse.

Corollary 5. *No selection rule that depends on the data only through the marginal law of $(C_1, \ldots, C_K, Z)$ can be simultaneously optimal for both composer classes.*

This covers accuracy ranking, pairwise diversity, error-correlation minimization, determinantal diversity on error vectors, greedy coverage, and greedy mutual information on labels, since all of them are functions of that law. It also explains the disagreement in the literature: a comparison run inside the regime of Theorem 3 concludes that homogeneous teams win (Li et al., 2026b), whereas one run outside it concludes that mixing families wins (Wang et al., 2025).

A natural reading of Theorem 3 is that one should measure Δ once on the full pool and let it pick the branch. This reading is wrong, and the reason is the budget.

Proposition 6 (A pool-level gap gives no budget-consistent branch rule)**.** *$\Delta(\cdot)$ is neither monotone nor submodular, and for every $\epsilon > 0$ there is an instance with $\Delta(\mathcal{M}) \le \epsilon$ on which*

$$\max_{|S|\le 2} V^{\dagger}(S) \;-\; \max_{|S|\le 2} F_Z(S) \;\ge\; 1/4. \tag{7}$$

Moreover, instances with the same value of $\Delta(\mathcal{M})$ can favor different branches under the budget, so no threshold rule on $\Delta(\mathcal{M})$ agrees with the budget-restricted branch comparison on all instances.

The mechanism is dilution. Redundant generalists shrink $\Delta(\mathcal{M})$ by making individual solvability common (Eq. (6)), while a small portfolio may still favor complementary specialists. Full-pool outcome-oracle certificates and all-wrong rates (Shihab et al., 2026; Chen, 2026) therefore need not reflect the gains available to a size-k portfolio. Together with Theorems 4 and 7, this motivates optimizing each branch under the budget and comparing the resulting portfolios.

4.3 Tractability is asymmetric

Theorem 7 (Hardness of the assembly branch)**.** *$V^{\dagger}$ is monotone but not submodular, and greedy applied to it can return 0.6 of the optimum, below the submodular bound $1-1/e \approx 0.632$. Moreover, unless $\mathrm{P} = \mathrm{NP}$, no polynomial-time algorithm approximates $\max_{|S|\le k} V^{\dagger}(S)$ within any constant factor: with deterministic competence vectors, $V^{\dagger}(S) = 1$ exactly when S_Φ covers every subgoal, so a constant-factor approximation decides set cover (Feige, 1998).*

Proposition 13 in Appendix A gives the witnessing instances. Hardness is the algorithmic content of the dichotomy. Coverage is easy because a portfolio's value is a union, and unions saturate. Assembly is an intersection over subgoals of unions over members, and an intersection is all or nothing. A coverage greedy therefore cannot simply be reused with a different score, which is why the two branches receive different algorithms. We instead obtain a guarantee on the logarithmic scale. With $g_\ell(S) = \max_{m\in S_\Phi} \nu_{m\ell}(\mathcal{H})$, define

$$\mathcal{L}(S) \;=\; \mathbb{E}\Big[\sum_{\ell=1}^{L} \log g_\ell(S)\Big]. \tag{8}$$

A product of per-subgoal terms suggests assuming that subgoal outcomes are conditionally independent. Under Eq. (2) this is false, since two graded competences of one member share every common ancestor. We therefore state the assumption one level down, where it is the standard local independence condition of item response theory (Lord, 1980) and where the low-rank model of Section 5 already imposes it.

Assumption 8 (Local independence and prefix support)**.** There is a latent per-subgoal difficulty vector D, determined by the query and hence $\mathcal{H}$-measurable, such that, conditional on $(\mathcal{H}, D)$, the intrinsic competences $\{A_{m\ell}\}$ are independent and the anchor satisfies $\alpha_{\Phi\ell}(\mathcal{H}, D) \ge \eta > 0$ almost surely, where $\alpha_{m\ell}(\mathcal{H}, D) = \mathbb{P}[A_{m\ell} = 1 \mid \mathcal{H}, D]$. In addition, for every m and ℓ used by the optimizer, $\nu_{mj}(\mathcal{H}, D) \ge \zeta > 0$ for all $j \in \mathrm{an}(\ell)$, either as observed or as supplied by a prefix-forced probe.

The assumption places no independence requirement on the observed competences, and the dependence induced by propagation actually helps. The graded events are increasing functions of an independent family, so Harris's inequality bounds their conjunction below by the product of their marginals (Lemma 14, Appendix A). The independence form of the surrogate therefore remains a valid lower bound once the assumption is stated at the intrinsic level, and the next theorem survives error propagation. Propagation damages tightness, not validity, and the damage is structured.

Theorem 9 (Submodular surrogate with an exponentiated certificate)**.** *Under Assumption 8, $S \mapsto \mathcal{L}(S) - \mathcal{L}(\emptyset)$ is normalized, monotone, and submodular, and $\log V_{\mathcal{H}}(S) \ge \mathcal{L}(S)$. Writing $L_0 = \exp \mathcal{L}(\emptyset) \le V_{\mathcal{H}}(\emptyset)$, greedy on $\mathcal{L}$ returns S^{gr} with*

$$\frac{V_{\mathcal{H}}(S^{\mathrm{gr}})}{L_0} \;\ge\; \left(\frac{\exp \mathcal{L}(S^{\star}_{\mathcal{L}})}{L_0}\right)^{1-1/e}, \qquad S^{\star}_{\mathcal{L}} = \arg\max_{|S|\le k} \mathcal{L}(S). \tag{9}$$

The guarantee is a $(1-1/e)$ factor on the log scale, and by Theorem 7 no polynomial-time constant-factor approximation of $V^{\dagger}$ itself exists unless P = NP. Because $\exp \mathcal{L}(S^{\mathrm{gr}})$ is computable and lower-bounds $V_{\mathcal{H}}(S^{\mathrm{gr}})$, Eq. (9) certifies the returned portfolio rather than claiming near-optimality.

4.4 CORRECTING THE SURROGATE FOR ERROR PROPAGATION

Propagation adds a second source of assembly advantage, with its own identity.

Proposition 10 (Propagation-recoverable mass)**.** *For every* $S \subseteq \mathcal{M}$, $V^{\ddagger}(S) - V^{\dagger}(S) = \mathbb{P}[\bigvee_{m\in S_\Phi} A_m = \mathbf{1} \wedge \bigvee_{m\in S_\Phi} B_m \neq \mathbf{1}] =: \Delta_{\text{prop}}(S) \geq 0$*, and consequently the total assembly advantage decomposes into two disjoint sources,*

$$V^{\ddagger}(S) - C(S) \;=\; \underbrace{\Delta(S)}_{\textit{distributed solvability}} \;+\; \underbrace{\Delta_{\text{prop}}(S)}_{\textit{propagation repair}} . \tag{10}$$

The two terms call for different interventions. Δ motivates retaining complementary specialists, which is a portfolio decision. Δ_{prop} captures subgoal competence blocked by upstream failures and motivates exposing and repairing intermediate results, which is an *architecture* decision.

Appendix B develops the corresponding correction to the surrogate. Under Assumption 8, ν is a product of intrinsic competences over a closed ancestor set, so these competences can be recovered in closed form by Möbius inversion wherever the prefix has positive support (Proposition 17). Substituting intrinsic for propagated competence in Eq. (8) yields a splice-capable surrogate $\mathcal{L}^{\ddagger}$. It remains monotone submodular and gives a uniformly tighter logarithmic lower bound than $\mathcal{L}$ (Theorem 18). The correction is not a constant shift: Proposition 19 bounds it below by a quantity that can grow quadratically with prerequisite depth. The uncorrected surrogate discounts member m's competence on subgoal ℓ by the product of its ancestor competences, which penalizes most the members that are weak early and strong late. Propagation bias thus compounds coverage bias against complementary specialists (Theorem 4; Corollary 5).

5 CAPS: COMPOSITION-AWARE PORTFOLIO SELECTION

Corollary 5 and Proposition 6 show that the branch cannot be read off the marginal law of correctness or a pool-level gap, and Theorem 7 shows that the two branches need different optimizers. CAPS therefore optimizes each branch separately and compares the results, in the four stages below.

Stage 1: sparse probing and subgoal tensor completion. Eqs. (3) and (4) need a tensor of size $K \times n \times L$, which cannot be filled by brute force when K reaches the thousands. Generations are stochastic, so we grade T draws per observed cell and target the success probability; grading a single draw would inflate every ceiling. We observe a random fraction ρ of cells and complete the rest with a multidimensional item-response model (Song et al., 2025; Zhuang et al., 2025),

$$\bar{\alpha}_{mi\ell} \;\approx\; \sigma(\langle u_m, v_{i\ell}\rangle + a_m + b_{i\ell}) , \qquad u_m, v_{i\ell} \in \mathbb{R}^r , \tag{11}$$

fit by regularized maximum likelihood on the observed entries. This reduces the probe requirement from $O(KnL)$ to $O((K + nL)r)$, and because u_m is a model-level embedding, a newly released candidate can be placed after a few probes. Eq. (11) also realizes Assumption 8 concretely, with the item factors playing the role of D. Section 4.4 dictates two further details. First, we fit at the intrinsic level: we read the prerequisite DAG off the benchmark, obtain the targets by the inversion in Eq. (15), and recover graded rates through $\log \bar{\nu}_{mi} = \Gamma \log \bar{\alpha}_{mi}$ (Proposition 17). Second, prefix support often fails for weak members on deep subgoals, which is exactly where Proposition 19 says the bias is largest. For such cells we issue a *prefix-forced probe*: we re-run the member on subgoal ℓ with correct ancestor outputs injected, which measures intrinsic competence directly.

Stage 2: identifiability heads. The deployable objectives need conditional rather than marginal quantities. We therefore fit $\widehat{\mu}_m(z)$ on a frozen text encoder (Reimers & Gurevych, 2019; Li et al., 2023), predicting all K correctness targets jointly so that no arbitrary single label is needed when several members are correct. Likewise, we fit $\widehat{\nu}_{m\ell}(h)$ on post-answer assembly features. Joint fitting matters because Eqs. (3) and (8) use the heads only through a maximum over members. Calibration *across* models therefore determines the objective value, whereas per-model accuracy does not.

Stage 3: branch enumeration and architecture diagnostics. Proposition 6 rules out choosing the branch from a pool-level gap. CAPS instead optimizes a portfolio for each available composer class and selects among the resulting portfolios by realized value on a held-out calibration split. This handles the operator–portfolio coupling of Theorem 4 explicitly. Under a cardinality budget and exact objective evaluations, the $(1 - 1/e)$ greedy guarantees apply to the normalized selection value and to the assembly log-surrogates (Theorems 1, 9, and 18). The added cost is small because all

branches share the graded tensor: each additional branch needs one greedy path and k composer calls for calibration. For composer classes not yet available, the gaps guide development priorities. In particular, Δ_{prop} quantifies the additional ceiling that propagation repair would unlock (Proposition 10). We estimate both gaps on labeled probes and report Clopper–Pearson upper limits U_Δ and U_{prop} at level $1-\delta$ (Clopper & Pearson, 1934). Comparing these limits with thresholds τ and τ_{prop} screens potential extensions. Appendix G.3 details their interpretation and the diagnostic shortcut.

Stage 4: debiased branch greedy. Both objectives are expectations of a maximum. The naive plug-in estimate is therefore optimistic by Jensen's inequality, and the optimism grows with $|S|$. This biases greedy, which compares sets of different cardinalities. We therefore select the maximizing member and evaluate its value on disjoint folds,

$$\widehat{F}^{\text{cf}}_Z(S) = \frac{1}{n}\sum_{i=1}^{n} \widehat{\mu}^{\,(f(i))}_{\widehat{\pi}_S^{(-f(i))}(z_i)}(z_i), \qquad \widehat{\pi}_S^{(-f)}(z) = \arg\max_{m \in S_\Phi} \widehat{\mu}_m^{(-f)}(z), \tag{12}$$

and proceed analogously inside Eqs. (8) and (16). The following proposition bounds the bias of Eq. (12) and shows how errors in the estimated marginal gains affect greedy selection.

Proposition 11 (Split-argmax debiasing and approximate greedy)**.** *Let ϵ bound $\mathbb{E}[\max_{m\in\mathcal{M}\cup\{\Phi\}} |\widehat{\mu}_m(Z) - \mu_m(Z)|]$ for every cross-fitted head. The estimator in Eq. (12) satisfies $|\mathbb{E}\widehat{F}^{\text{cf}}_Z(S) - F_Z(S)| \le 3\epsilon$ for every S, a bound independent of $|S|$, whereas the naive plug-in is optimistic by an amount of order $\sigma\sqrt{2\log|S|}$ when the head errors are independent across models with scale σ. If marginal gains are evaluated to within ϵ' uniformly over the sets queried, greedy returns S^{gr} with $F_Z(S^{\text{gr}}) \ge (1-1/e)\max_{|S|\le k} F_Z(S) - 2k\epsilon'$.*

Greedy produces a nested sequence of candidate portfolios, and calibration decides which prefix to deploy. Real assembly accuracy can fall as conflicting responses accumulate, even though the surrogate is monotone. We therefore run the actual composer at each prefix on held-out calibration data, fit a one-dimensional attenuation curve over portfolio size, and deploy the prefix that maximizes the attenuated objective. The number of calibrated portfolios thus stays linear in k.

6 Experiments

6.1 Experiments Setup

Datasets. We use six benchmarks with observable subgoals spanning both regimes of Theorem 3: **LiveCodeBench** (Jain et al., 2025) and **EvalPlus** (Liu et al., 2023) provide per-test pass vectors, making Eq. (1) exact; **MuSiQue** (Trivedi et al., 2022) and **HotpotQA** (Yang et al., 2018) provide annotated sub-questions; **MATH-500** (Hendrycks et al., 2021) uses step-level grading following Lightman et al. (2024); and **MMLU-Pro** (Wang et al., 2024) is the atomic control with coinciding ceilings. Large-pool selection-only experiments use correctness matrices from **RouterBench** (Hu et al., 2024), **LLMRouterBench** (Li et al., 2026a), and **RouterEval** (Huang et al., 2025), which lack subgoal labels, to evaluate coverage and completion at K up to several thousand. The live pool contains 42 checkpoints from nine families, ranging from 0.5B to 72B parameters (Appendix F). Each dataset has separate probe and test splits, with the test split opened once.

Baselines. We compare against every family Corollary 5 covers, plus the strongest composition systems: best single model, cheapest model, random k, top-k accuracy, minimum pairwise error correlation, determinantal diversity on error vectors (Kulesza & Taskar, 2012), greedy complementary coverage (Shihab et al., 2026), greedy mutual information (Turkmen et al., 2026), approximate Shapley values (Shapley et al., 1953), LLM-as-a-judge scoring (Zheng et al., 2023), Self-MoA (Li et al., 2026b), Mixture-of-Agents with per-family top-1 (Wang et al., 2025), the greedy variants of Zhang et al. (2026), and Archon inference-time architecture search (Saad-Falcon et al., 2024).

Metrics and protocol. We report the ladder in Eq. (10), with 95% upper confidence limits for the two **gaps**, alongside **realized accuracy** on the untouched test split. **Cost** combines normalized token spend and measured latency and is reported as a frontier; **stability** is the mean pairwise Jaccard overlap of selected portfolios across folds. All correctness quantities use $T = 5$ at temperature 0.7.

6.2 Results and Analysis

The decomposability gap tracks assembly gains. Table 1 reports the ceilings and gaps. On atomic MMLU-Pro ($L = 1$), the estimated decomposability gap is 0.021 (upper limit 0.041), with statis-

Table 1: The ladder of Eq. (10) at $k = 4$ on the 42-model pool, with the two gaps that decompose it and their 95 percent upper limits. $\bar{L}$ is the mean number of subgoals, $\bar{d}$ the mean prerequisite depth, and Branch the composer class selected by the diagnostic.

		Ladder			Gaps [95% UL]		
Benchmark	$\bar{L}/\bar{d}$	C	$V^{\dagger}$	$V^{\ddagger}$	$\widehat{\Delta}$	$\widehat{\Delta}_{\text{prop}}$	Branch
MMLU-Pro	1.0 / 0.0	.871	.892	.896	.021 [.041]	.004 [.014]	coverage
MATH-500	6.8 / 5.8	.804	.898	.961	.094 [.121]	.063 [.094]	splice
HotpotQA	2.4 / 1.4	.762	.930	.964	.168 [.199]	.034 [.059]	splice
MuSiQue	3.1 / 2.0	.684	.897	.953	.213 [.248]	.056 [.089]	splice
EvalPlus	14.2 / 1.0	.793	.980	.992	.187 [.216]	.012 [.029]	splice-free
LiveCodeBench	21.6 / 1.0	.611	.852	.944	.241 [.279]	.092 [.127]	splice

Table 2: Realized accuracy at $k = 4$, mean over 5 random seeds with standard deviation, under the branch-appropriate composer. Non-deployable references in the last block.

Method	MMLU-Pro	MATH-500	HotpotQA	MuSiQue	EvalPlus	LCB
Best single model	.662 $\pm$.000	.571 $\pm$.000	.498 $\pm$.000	.391 $\pm$.000	.604 $\pm$.000	.372 $\pm$.000
Random k	.618 $\pm$.019	.533 $\pm$.024	.471 $\pm$.022	.368 $\pm$.026	.572 $\pm$.023	.341 $\pm$.028
Top-k accuracy	.671 $\pm$.006	.584 $\pm$.008	.509 $\pm$.009	.402 $\pm$.011	.617 $\pm$.008	.385 $\pm$.012
Min error correlation	.659 $\pm$.008	.579 $\pm$.010	.515 $\pm$.010	.414 $\pm$.013	.621 $\pm$.010	.391 $\pm$.014
DPP on error vectors	.663 $\pm$.009	.581 $\pm$.011	.518 $\pm$.011	.417 $\pm$.012	.624 $\pm$.011	.394 $\pm$.013
Greedy coverage	.679 $\pm$.005	.592 $\pm$.007	.521 $\pm$.008	.413 $\pm$.010	.628 $\pm$.008	.396 $\pm$.011
Greedy mutual info	.674 $\pm$.006	.589 $\pm$.008	.524 $\pm$.009	.420 $\pm$.011	.631 $\pm$.009	.401 $\pm$.012
Approx. Shapley	.670 $\pm$.010	.586 $\pm$.013	.519 $\pm$.014	.411 $\pm$.016	.626 $\pm$.013	.393 $\pm$.017
LLM judge	.651 $\pm$.012	.563 $\pm$.015	.494 $\pm$.016	.386 $\pm$.018	.599 $\pm$.015	.368 $\pm$.019
Self-MoA	.676 $\pm$.007	.588 $\pm$.009	.506 $\pm$.010	.397 $\pm$.012	.615 $\pm$.010	.379 $\pm$.013
MoA per-family top-1	.668 $\pm$.008	.590 $\pm$.010	.528 $\pm$.010	.424 $\pm$.013	.637 $\pm$.010	.408 $\pm$.014
Model-first greedy	.681 $\pm$.006	.597 $\pm$.008	.531 $\pm$.009	.428 $\pm$.011	.641 $\pm$.009	.412 $\pm$.012
Archon ITAS	.677 $\pm$.009	.594 $\pm$.011	.529 $\pm$.012	.426 $\pm$.014	.639 $\pm$.011	.410 $\pm$.015
CAPS (ours)	**.685** $\pm$.005	**.618** $\pm$.007	**.559** $\pm$.008	**.463** $\pm$.009	**.673** $\pm$.007	**.461** $\pm$.010
Non-deployable references						
Coverage ceiling C	.871	.804	.762	.684	.793	.611
Splice-free ceiling $V^{\dagger}$	.892	.898	.930	.897	.980	.852
Splice-capable ceiling $V^{\ddagger}$	.896	.961	.964	.953	.992	.944
Exhaustive optimum	.688	.624	.566	.471	.681	.470

tically indistinguishable ceilings, which is consistent with Theorem 3. On LiveCodeBench, the gap reaches 0.241, reflecting queries solved collectively but by no individual member. Across benchmark–pool combinations, larger gaps are associated with a greater advantage for assembly portfolios over coverage portfolios, as shown in Figure 2 and further analyzed in Appendix E.1.

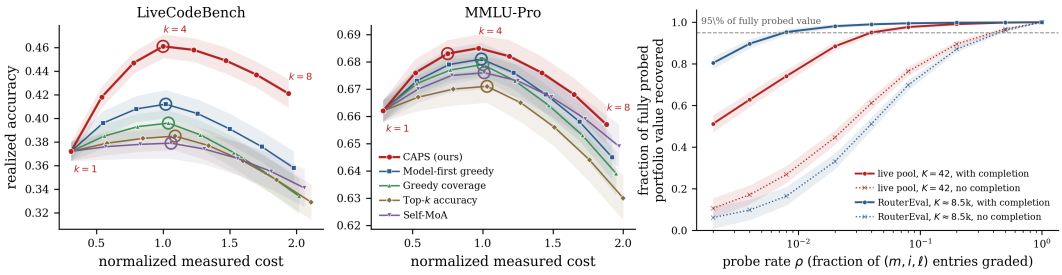
The two gaps imply different interventions. EvalPlus has more subgoals than MATH-500 but shallower prerequisites (1.0 versus 5.8 mean depth) and a smaller propagation gap, consistent with the role of depth in Eq. (17). Eq. (10) thus separates gains from portfolio complementarity on EvalPlus from the additional gains that intermediate-result repair could bring on MATH-500 and LiveCodeBench, a distinction that Δ alone misses. Using graded rather than deconvolved competence costs 1.9 percentage points and reduces the selected share of specialists from 0.46 to 0.18 (Table 4, Appendix E), consistent with Proposition 19. Specialists are members below the pool median in overall accuracy but above the third quartile on some intrinsic subgoal rate. Prefix-forced probes account for 0.8 of these points and target the 7.4% of cells with insufficient prefix support.

Composition-aware selection improves realized accuracy. Table 2 reports realized accuracy under the composer that each benchmark's diagnostic selects, and CAPS is best on all six benchmarks. Its margin over the strongest prior rule is 0.4 points on MMLU-Pro, where Theorem 3 predicts no advantage, and 4.9 points on LiveCodeBench, where the gap is largest. The baseline ranking also shifts with the regime, which is the practical form of Corollary 5. Greedy coverage is second only to model-first greedy on MMLU-Pro but ranks seventh on MuSiQue. Self-MoA is competitive on MMLU-Pro but clearly dominated once the gap exceeds 0.15. Top-k accuracy and the diversity rules are never best anywhere, consistent with all three being functions of the marginal law alone.

Branch enumeration preserves gains under dilution. Table 3 evaluates each objective under each composer. Across the four high-gap benchmarks, a mismatch between objective and composer

Table 3: Branch ablation. The upper block fixes the selection objective and varies the deployed composer; the lower block varies how the branch is chosen. Realized accuracy averaged over the four benchmarks with $\widehat{\Delta} > 0.15$, and over MMLU-Pro alone in the last column.

Selection objective	Selection composer	Assembly composer	MMLU-Pro (atomic)
Coverage ceiling C	.481 $\pm$.009	.501 $\pm$.010	.679 $\pm$.005
Deployable selection $\widehat{F}_Z$ (ours)	**.518** $\pm$.008	.513 $\pm$.009	**.685** $\pm$.005
Assembly surrogate $\widehat{\mathcal{L}}$ (ours)	.492 $\pm$.010	**.539** $\pm$.008	.682 $\pm$.006
Oracle branch choice	.541 $\pm$.008		.685 $\pm$.005
CAPS enumerate and compare	**.540** $\pm$.008		**.685** $\pm$.005
Diagnostic as a shortcut, original pool	.539 $\pm$.008		.685 $\pm$.005
Diagnostic as a shortcut, augmented pool	.521 $\pm$.011		.684 $\pm$.006
Random branch choice	.516 $\pm$.014		.681 $\pm$.007



(a) Cost and accuracy frontier.

(b) Probe efficiency.

Figure 1: (a) Cost–accuracy frontiers across portfolio sizes k. (b) Portfolio value recovered by low-rank completion at different probe rates ρ.

costs 3.8 points on average for coverage under assembly and 2.6 points for the assembly surrogate under selection, relative to the matching objective (Theorem 4). Enumerating the three classes and comparing their portfolios on calibration data recovers 98% of the oracle branch choice's gain over greedy coverage, versus 51% for random choice, at the cost of three greedy paths and $3k$ rather than k composer calls. The diagnostic shortcut recovers 96% on the original pool but is sensitive to dilution. Adding six near-duplicate checkpoints from existing families lowers the pool-level U_Δ from 0.199 to 0.047; the shortcut then selects coverage on four of six benchmarks and loses 1.9 points, whereas enumeration is unaffected. Consistent with Proposition 6, pool-level gaps can guide architecture investment but cannot reliably determine the best branch under a portfolio budget.

Calibration and completion improve cost efficiency. Figure 1a sweeps k from 1 to 8 against measured cost. CAPS dominates the frontier throughout, and the curves reveal what the fixed-k tables cannot. Coverage baselines keep adding members past the point where the composer degrades, so their curves turn down after $k = 5$, whereas the attenuation calibration stops CAPS between 3 and 5. Figure 1b varies the probe rate: low-rank completion recovers 95 percent of the fully probed portfolio value at $\rho = 0.04$ on the live pool, and at $\rho = 0.008$ on RouterEval, where K is in the thousands and the latent factors are shared across many near-duplicate checkpoints. This makes the method usable where exhaustive grading is out of reach.

7 Conclusions and Future Work

Model portfolio selection has been posed without reference to the operator that consumes the portfolio. Making this operator explicit separates selection from assembly, yielding distinct value functionals, structures, and complexity. Two estimable probabilities guide portfolio and architecture decisions, respectively. These results bring two practical implications. First, correcting the mismatch between grader reports and objective-relevant quantities preserves complementary specialists. Second, pool-level gaps may be unattainable under budget, so they should guide architecture design rather than portfolio selection. Our framework has four main limitations: conjunctive grading assumptions, metadata-derived prerequisites, static portfolios, and no clear analogue to prefix-forced probing for open-ended generation. Future work includes continual rebalancing under switching costs, robust selection over task mixtures, and budgeted probe identification.

AI USE STATEMENT

In this work, we used generative AI tools only to edit the manuscript for readability, specifically for grammar and spelling correction, rephrasing individual sentences for concision, and smoothing transitions between paragraphs. We have not used generative AI tools for any task carrying a required disclosure: the problem formulation, the composer dichotomy and its value functionals, the statements and proofs of all theorems, propositions, and lemmas, the design of CAPS and of the evaluation protocol, the implementation of the method and of every baseline, the probing and grading pipeline, and the interpretation of the reported results are the authors' own work. No synthetic data were generated with AI assistance, no dataset was cleaned or reformatted by an AI tool, and no translation was involved. Tasks in the recommended category other than readability editing, including literature search and identification, figure creation, code writing, and reference formatting, were likewise carried out by the authors without AI assistance. Every AI-suggested edit was reviewed by at least one author against the original text and accepted only after confirming that no claim, number, citation, or technical statement had been altered, added, or removed, and all authors have read the final manuscript in full. We take responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

The authors have read and adhere to the ICLR Code of Ethics. This work involves no human subjects, no user data, and no collection or release of new datasets. All experiments use publicly available benchmarks (MMLU-Pro, MATH-500, HotpotQA, MuSiQue, EvalPlus, LiveCodeBench) and publicly released correctness matrices (RouterBench, LLMRouterBench, RouterEval), together with 42 openly licensed instruction-tuned checkpoints, each used in accordance with its license, and we release only our own code and derived grade tensors rather than redistributing model weights or benchmark content. Model-generated programs for the two code suites are executed in a network-isolated container under per-test time and memory limits, since running untrusted generated code is the principal operational risk in our pipeline. Two limitations are worth stating as ethical rather than merely technical. First, our objectives score a portfolio by correctness and serving cost on a specified task distribution, so a portfolio that is optimal in our sense carries no guarantee about safety, toxicity, or fairness, and a deployment adopting this procedure should impose such criteria as separate constraints rather than assume they follow from accuracy. Second, the probe distribution defines what counts as competence, so a probe set that underrepresents a language, domain, or user population can lead the selector to discard exactly the members that serve it, an effect consistent with the distribution shift results of Table 5 and a reason we recommend re-selection when the task mixture moves. We report total compute in Appendix G.6 for transparency about the energy cost of the study. We declare no conflicts of interest or sponsorship that could have influenced the reported results; test splits were opened once after all hyperparameters were fixed on the probe split, and every reported number is emitted from stored per-item records by a single script.

REPRODUCIBILITY STATEMENT

We have tried to make every claim in this paper independently checkable. On the theoretical side, the single structural assumption is stated in Eq. 1 and refined by Eq. 2, the local independence and prefix support conditions are stated in full as Assumption 8, the statements deferred from the main text appear in Appendices A and B, and complete proofs of every lemma, proposition, theorem, and corollary are given in Appendix C, with the separating instances of Theorem 4, Proposition 6, Theorem 7, and Proposition 13 written out explicitly so that each can be verified by hand. On the empirical side, Appendix F lists all 42 checkpoints of the live pool with truncated commit hashes resolved against the revision file, measured latencies on both inference paths, and the normalized cost weights used under a knapsack budget, together with the by-family split used for the held-out-model study. Appendix G documents the rest of the pipeline: generation, grading, and the multi-draw protocol in G.1, the subgoal units, graders, prerequisite DAG shapes, and prefix-forced probing rules in G.2 and Table 7, the complete hyperparameter table with how each value was selected in G.3 and Table 8, the three composer configurations in G.4, every prompt template in full in G.5, and the baseline budgets and total compute in G.6. All benchmarks and router correctness matrices we

use are public and cited in Section 6.1. The code is available at https://anonymous.4open.science/r/CAPS-3AB8, and Appendix G.1 notes the precision at which stored confidences must be kept, since rounding them is by itself enough to move a confidence-gated baseline.

REFERENCES

Josef Chen. When does combining language models help? a co-failure ceiling on routing, voting, and mixture-of-agents across 67 frontier models. *arXiv preprint arXiv:2606.27288*, 2026.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4): 634–652, 1998.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 3860–3887. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-EMNLP.208. URL https://doi.org/10.18653/v1/2025.findings-emnlp.208.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3(71-104): 3, 2014.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, et al. Llmrouterbench: A massive benchmark and unified framework for llm routing. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37733–37754, 2026a.

Wenzhe Li, Yong Lin, Mengzhou Xia, and Chi Jin. Rethinking mixture-of-agents: Is mixing different large language models beneficial? *Trans. Mach. Learn. Res.*, 2026, 2026b. URL https://openreview.net/forum?id=K6WwK8URlV.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Frederic M Lord. *Applications of item response theory to practical testing problems*. Routledge, 1980.

Yifan Lu, Qiyue Zhang, Shenrun Zhang, Zhibo Yu, Zhuang Wang, Hanjie Chen, and Jiarong Xing. The routing plateau: Understanding and breaking the accuracy limits of llm routers. *arXiv preprint arXiv:2606.07587*, 2026.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

John R Rice. The algorithm selection problem. In *Advances in computers*, volume 15, pp. 65–118. Elsevier, 1976.

Jon Saad-Falcon, Adrian Gamarra Lafuente, Shlok Natarajan, Nahum Maru, Hristo Todorov, Etash Guha, E Kelly Buchanan, Mayee Chen, Neel Guha, Christopher Ré, et al. Archon: An architecture search framework for inference-time techniques. *arXiv preprint arXiv:2409.15254*, 2024.

Lloyd S Shapley et al. A value for n-person games. 1953.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Ibne Farabi Shihab, Abu Sa-Adat Mohamed Moon-Im Al Ahsan, and Md Najmus Swaqeeb. Opportunity is not realizability: Selection-valid diagnostics for multi-llm routing. *CoRR*, abs/2608.08265, 2026. doi: 10.48550/ARXIV.2608.08265. URL https://doi.org/10.48550/arXiv.2608.08265.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. Irt-router: Effective and interpretable multi-llm routing via item response theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15629–15644, 2025.

Kv Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. Harnessing the power of multiple minds: Lessons learned from llm routing. In *Proceedings of the Fifth Workshop on Insights from Negative Results in NLP*, pp. 124–134, 2024.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Yigit Turkmen, Baturalp Buyukates, and Melih Bastopcu. Don't always pick the highest-performing model: An information theoretic view of llm ensemble selection. *arXiv preprint arXiv:2602.08003*, 2026.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Satzilla: portfolio-based algorithm selection for sat. *Journal of artificial intelligence research*, 32:565–606, 2008.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025.

Yichi Zhang, Kevin Lu, Yuang Zhang, Jie Gao, Lirong Xia, and Fang-Yi Yu. Mixture of complementary agents for robust llm ensemble. *arXiv preprint arXiv:2605.24048*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. Embedllm: Learning compact representations of large language models. In *International Conference on Learning Representations*, volume 2025, pp. 76913–76926, 2025.

CONTENTS

1 Introduction **1**

2 Related Work **2**

3 Preliminaries **3**

4 The Two Branches Are Different Set Functions **4**

4.1 Selection is submodular at every information level 4

4.2 Assembly: an exact identity and a dichotomy 4

4.3 Tractability is asymmetric 5

4.4 Correcting the surrogate for error propagation 6

5 CAPS: Composition-Aware Portfolio Selection **6**

6 Experiments **7**

6.1 Experiments Setup 7

6.2 Results and Analysis 7

7 Conclusions and Future Work **9**

A Deferred Statements **17**

B The Propagation-Corrected Surrogate **17**

C Proofs **18**

C.1 Proof of Lemma 12 18

C.2 Proof of Theorem 1 18

C.3 Proof of Proposition 2 18

C.4 Proof of Theorem 3 18

C.5 Proof of Theorem 4 19

C.6 Proof of Corollary 5 19

C.7 Proof of Proposition 6 19

C.8 Proof of Proposition 13 and Theorem 7 20

C.9 Proof of Lemma 14 20

C.10 Proof of Theorem 9 20

C.11 Proof of Proposition 10 21

C.12 Proof of Proposition 17 21

C.13 Proof of Theorem 18 21

C.14 Proof of Proposition 19 21

C.15 Proof of Proposition 11 22

D Pseudocode **22**

E Additional Experimental Results **23**

E.1 Diagnostic validation 23

E.2 Component ablation 23

E.3 Generalization and stability 24

F Model Pool **25**

G Implementation Details **25**

G.1 Generation, grading, and stochasticity 25

G.2 Subgoal extraction and prerequisite DAGs 25

G.3 Estimators, optimizer, and thresholds 27

G.4 Composer configurations 27

G.5 Prompt templates 28

G.6 Baselines and compute 29

H Extended Limitations and Future Work **29**

A Deferred Statements

This appendix formally states the three results that the main text summarizes without display, and adds two remarks on the residual looseness of the splice-free branch. All proofs are in Appendix C.

Lemma 12 (Information ladder). *If $\mathcal{G}_1 \subseteq \mathcal{G}_2$ then $F_{\mathcal{G}_1}(S) \le F_{\mathcal{G}_2}(S)$ for every S, and likewise for V. Consequently $\max_{m\in S_\Phi} \mathbb{E}C_m \le F_Z(S) \le C(S)$ and $V_{\mathcal{H}}(S) \le V^\dagger(S) \le V^\ddagger(S)$.*

Proposition 13 (Structure of the assembly ceiling). *$V^\dagger$ is monotone but not submodular. In the instance of Theorem 4, $V^\dagger(\emptyset) = V^\dagger(\{s_1\}) = V^\dagger(\{s_2\}) = 0$ while $V^\dagger(\{s_1, s_2\}) = 1$, so the marginal gain of s_2 strictly increases when s_1 is added. If the latent success probability of the generalists is lowered to 0.6, greedy on $V^\dagger$ at $k = 2$ terminates at value 0.6 while the optimum is 1, a ratio strictly below $1 - 1/e$.*

Lemma 14 (Propagation is positively associated). *Under Assumption 8, for any assignment π measurable with respect to $(\mathcal{H}, D)$,*

$$\mathbb{P}\Big[\forall \ell : B_{\pi(\ell)\ell} = 1 \;\Big|\; \mathcal{H}, D\Big] \;\ge\; \prod_{\ell=1}^{L} \nu_{\pi(\ell)\ell}(\mathcal{H}, D), \tag{13}$$

and the same holds with B replaced by A. The inequality is an equality when G has no edges.

Remark 15 (The residual propagation surplus). Write $Y_\ell(S) = \max_{m\in S_\Phi} B_{m\ell}$ and condition on $(\mathcal{H}, D)$ throughout. The chain rule in topological order gives

$$\log \mathbb{P}\big[\forall \ell : Y_\ell(S) = 1\big] = \sum_{\ell=1}^{L} \log \mathbb{P}\big[Y_\ell(S) = 1 \,\big|\, Y_j(S) = 1 \;\forall j \prec \ell\big] = \mathcal{L}_{\mathrm{ind}}(S) + \mathrm{TC}(S), \tag{14}$$

where $\mathcal{L}_{\mathrm{ind}}(S) = \sum_\ell \log \mathbb{P}[Y_\ell(S) = 1] \ge \sum_\ell \log g_\ell(S)$ collects the unconditional terms. The first equality needs no assumption, and $\mathrm{TC}(S) \ge 0$ by Harris's inequality as in Lemma 14, since each $Y_\ell(S)$ is increasing in the intrinsic competences. Equation (14) is exact, but its terms are not submodular because the conditioning event itself depends on S. Discarding the surplus is therefore the price of submodularity, and Lemma 14 certifies that the price is paid in the conservative direction.

Remark 16 (Depth dependence of the certificate). Assumption 8 bounds the anchor competence below by η, so $L_0^\ddagger \ge \eta^L$, while the splice-free constant satisfies $L_0 \ge \eta^{\sum_\ell |\mathrm{an}^*(\ell)|}$. The certificate therefore degrades with prerequisite depth in the splice-free branch but only with subgoal count in the splice-capable one, which is a second reason to prefer a splice-capable composer when the propagation gap is large.

B The Propagation-Corrected Surrogate

This appendix contains the results summarized in Section 4.4. Throughout, $\alpha_{m\ell}(\mathcal{H}, D) = \mathbb{P}[A_{m\ell} = 1 \mid \mathcal{H}, D]$ is the intrinsic competence of Eq. (2), and Assumption 8 is in force. Proofs are in Appendix C.

Proposition 17 (Propagation deconvolution). *Let $\Gamma \in \{0,1\}^{L\times L}$ be the closed reachability matrix of G, $\Gamma_{\ell j} = \mathbb{1}\{j \in \mathrm{an}^*(\ell)\}$. Under Assumption 8, $\log \nu_m = \Gamma \log \alpha_m$ componentwise, and Γ is unit lower triangular in any topological order, hence invertible with integer inverse. Therefore*

$$\log \alpha_m \;=\; \Gamma^{-1} \log \nu_m, \tag{15}$$

the Möbius inversion of $\log \nu_m$ over the reachability order. For a chain this reduces to prefix-conditional ratios, $\alpha_{m\ell} = \nu_{m\ell}/\nu_{m,\ell-1}$.

Replacing propagated with intrinsic competence defines the splice-capable surrogate

$$\mathcal{L}^\ddagger(S) \;=\; \mathbb{E}\Big[\sum_{\ell=1}^{L} \log \max_{m\in S_\Phi} \alpha_{m\ell}(\mathcal{H}, D)\Big]. \tag{16}$$

Theorem 18 (Propagation-corrected surrogate). *Under Assumption 8, $\mathcal{L}^\ddagger - \mathcal{L}^\ddagger(\emptyset)$ is normalized, monotone, and submodular, and $\mathcal{L}(S) \le \mathcal{L}^\ddagger(S) \le \log V_{\mathcal{H}}^\ddagger(S)$ for every S, so greedy on $\mathcal{L}^\ddagger$ enjoys the analogue of Eq. (9) with $L_0^\ddagger = \exp \mathcal{L}^\ddagger(\emptyset)$. The left inequality is strict whenever, with positive probability, the propagated and intrinsic maximizers differ at some subgoal, which requires that subgoal to have an ancestor in G.*

Proposition 19 (Scale and direction of the propagation bias)**.** *Let $m^\star_\ell$ maximize $\nu_{\cdot\ell}$. Then*

$$\mathcal{L}^\ddagger(S) - \mathcal{L}(S) \;\geq\; \mathbb{E}\Big[\sum_{\ell=1}^{L} \sum_{j\in \mathrm{an}(\ell)} \log \frac{1}{\alpha_{m^\star_\ell j}(\mathcal{H}, D)}\Big] \;\geq\; 0, \tag{17}$$

with equality at zero when G has no edges. On a chain of length L with uniform intrinsic competence α the bound is $\binom{L}{2}\log(1/\alpha)$, so the understatement grows quadratically in prerequisite depth.

The undervaluation factor of member m at subgoal ℓ is exactly $\prod_{j\in \mathrm{an}(\ell)} \alpha_{mj}$. In the instance of Theorem 4 extended to a chain, this factor is zero for the terminal specialist, so the uncorrected surrogate assigns it no marginal value at any prefix and greedy never selects it. Remarks 15 and 16 in Appendix A give the residual looseness of the splice-free branch and the depth dependence of the certificate.

C Proofs

C.1 Proof of Lemma 12

Let $\mathcal{G}_1 \subseteq \mathcal{G}_2$. For each m, $\mathbb{E}[C_m \mid \mathcal{G}_2] \leq \max_{m'\in S_\Phi} \mathbb{E}[C_{m'} \mid \mathcal{G}_2]$ pointwise, and conditional expectation is monotone, so

$$\mathbb{E}[C_m \mid \mathcal{G}_1] = \mathbb{E}\big[\mathbb{E}[C_m \mid \mathcal{G}_2] \,\big|\, \mathcal{G}_1\big] \leq \mathbb{E}\Big[\max_{m'\in S_\Phi} \mathbb{E}[C_{m'} \mid \mathcal{G}_2] \,\Big|\, \mathcal{G}_1\Big].$$

The right side does not depend on m, so it also dominates the maximum over m of the left side. Taking expectations gives $F_{\mathcal{G}_1}(S) \leq F_{\mathcal{G}_2}(S)$. For the named consequences, take $\mathcal{G}_1$ trivial and $\mathcal{G}_2 = \sigma(Z)$ to get $\max_{m\in S_\Phi} \mathbb{E}C_m \leq F_Z(S)$. Then take $\mathcal{G}_1 = \sigma(Z)$ and $\mathcal{G}_2 = \mathcal{G}_\star := \sigma(Z, C_m : m \in \mathcal{M}\cup\{\Phi\})$, the full outcome information. Each C_m is $\mathcal{G}_\star$-measurable, so $\mathbb{E}[C_m \mid \mathcal{G}_\star] = C_m$ and hence $F_{\mathcal{G}_\star}(S) = \mathbb{E}[\max_{m\in S_\Phi} C_m] = C(S)$. The argument for V is identical, with $\prod_\ell B_{\pi(\ell)\ell}$ in place of C_m and the maximum over assignments π; taking $\mathcal{H}_2 = \mathcal{H} \vee \sigma(B_{m\ell} : m, \ell)$ gives $V_{\mathcal{H}}(S) \leq V^\dagger(S)$. Finally, $V^\dagger(S) \leq V^\ddagger(S)$ holds pointwise because $B_{m\ell} \leq A_{m\ell}$ by Eq. (2). □

C.2 Proof of Theorem 1

Fix an outcome ω and write $w_m = \mathbb{E}[C_m \mid \mathcal{G}](\omega) \in [0,1]$, so the integrand of Eq. (3) is $h_\omega(S) = \max_{m\in S_\Phi} w_m$. Monotonicity is immediate. For submodularity, let $S \subseteq T$ and $j \notin T$. Then

$$h_\omega(S\cup\{j\}) - h_\omega(S) = \Big(w_j - \max_{m\in S_\Phi} w_m\Big)^+. \tag{18}$$

Since $\max_{m\in S_\Phi} w_m \leq \max_{m\in T_\Phi} w_m$ and $t \mapsto (w_j - t)^+$ is nonincreasing, the marginal gain at S is at least that at T, so h_ω has diminishing returns. Normalization follows by subtracting $h_\omega(\emptyset) = w_\Phi$. All three properties are preserved by integration against a nonnegative measure, so $F_{\mathcal{G}} - F_{\mathcal{G}}(\emptyset)$ is normalized, monotone, and submodular. The cardinality guarantee then follows from Nemhauser et al. (1978) and the knapsack guarantee from Sviridenko (2004). □

Remark 20. Eq. (18) is where the two branches part company. The selection marginal is a truncated difference against a running maximum, which saturates. The assembly marginal is a difference of products of running maxima, which does not.

C.3 Proof of Proposition 2

Both quantities are expectations of indicators. By Eq. (1), $\max_{m\in S_\Phi} C_m = \mathbb{1}\{\exists m \in S_\Phi : B_m = \mathbf{1}\}$, and by definition $\prod_\ell \max_{m\in S_\Phi} B_{m\ell} = \mathbb{1}\{\bigvee_{m\in S_\Phi} B_m = \mathbf{1}\}$. The event $E_1 = \{\exists m : B_m = \mathbf{1}\}$ is contained in $E_2 = \{\bigvee_m B_m = \mathbf{1}\}$, so pointwise $\mathbb{1}_{E_2} - \mathbb{1}_{E_1} = \mathbb{1}_{E_2\setminus E_1}$. Taking expectations gives Eq. (6), and nonnegativity follows from the inclusion. □

C.4 Proof of Theorem 3

By Proposition 2, $V^\dagger(S) = C(S)$ if and only if $\mathbb{P}[E_2 \setminus E_1] = 0$, which for all S simultaneously is exactly the stated implication holding almost surely.

Nestedness is sufficient. If $\{B_m\}_{m \in S_\Phi}$ is a chain under the componentwise order, let $m^\top$ attain the maximum. Then $\bigvee_m B_m = B_{m^\top}$, so membership in E_2 forces $B_{m^\top} = \mathbf{1}$ and hence E_1.

The condition is strictly weaker. Take $L = 3$ and $S_\Phi = \{\Phi, 1, 2\}$ with $B_\Phi = (0,0,0)$, $B_1 = (1,0,0)$, and $B_2 = (0,1,0)$. This family is not a chain, yet $\bigvee_m B_m = (1,1,0) \neq \mathbf{1}$, so E_2 is empty and the implication holds vacuously. Finally, if $L = 1$ then $B_m = \mathbf{1}$ is the same event as $B_{m1} = 1$, and E_2 says that some $B_{m1} = 1$, so $E_2 = E_1$. □

C.5 Proof of Theorem 4

Take $L = 2$, $k = 2$, $\mathcal{M} = \{g_1, g_2, s_1, s_2\}$, anchor competence $B_\Phi = (0,0)$ almost surely, and a single latent $U \sim \text{Bernoulli}(0.8)$ revealed by the pre-answer signal, $Z = U$. Set $B_{g_1} = B_{g_2} = (U, U)$ and, deterministically, $B_{s_1} = (1,0)$ and $B_{s_2} = (0,1)$. By Eq. (1), $C_{g_1} = C_{g_2} = U$ and $C_{s_1} = C_{s_2} = 0$, so $\mu_{g_i}(Z) = U$ and $\mu_{s_i}(Z) = \mu_\Phi(Z) = 0$.

Selection values. For any pair containing a generalist, $F_Z(S) = \mathbb{E}[\max(0, U)] = 0.8$, and this is the maximum since $\mu_m \leq U$ for every m. For the specialist pair, $F_Z(\{s_1, s_2\}) = \mathbb{E}[\max(0,0,0)] = 0$. Hence $\arg\max_{|S|=2} F_Z$ is the collection of pairs containing at least one generalist, and $\{s_1, s_2\}$ is not in it.

Assembly values. $V^\dagger(\{s_1, s_2\}) = \mathbb{E}[\max(0,1,0) \cdot \max(0,0,1)] = 1$, while $V^\dagger(\{g_1, g_2\}) = \mathbb{E}[U \cdot U] = 0.8$ and $V^\dagger(\{g_i, s_j\}) = \mathbb{E}[1 \cdot U] = 0.8$. Hence $\arg\max_{|S|=2} V^\dagger = \{\{s_1, s_2\}\}$ uniquely.

The two argmax sets are therefore disjoint. Deploying a selection-optimal portfolio under an assembly composer yields 0.8 against an available 1, a loss of 0.2; deploying the assembly-optimal portfolio under a selection composer yields 0 against an available 0.8, a loss of 0.8. □

C.6 Proof of Corollary 5

We exhibit two instances with the same joint law of $(C_1, \ldots, C_K, Z)$ but different assembly optima. Instance one is the construction of Theorem 4. Instance two replaces the specialists by s_1' and s_2' with $B_{s_i'} = (0,0)$ almost surely, leaving the generalists and Z unchanged. In both instances $C_{g_1} = C_{g_2} = U$ and the remaining two correctness variables are identically zero, so the joint laws coincide, and any rule that is a measurable function of this law returns the same portfolio in both. However, $V^\dagger(\{s_1, s_2\}) = 1$ in instance one, while $V^\dagger(\{s_1', s_2'\}) = 0$ in instance two, where the assembly optimum is any pair containing a generalist, with value 0.8. No single rule can be assembly-optimal in both, and in instance one no rule can be simultaneously selection-optimal and assembly-optimal by Theorem 4. □

C.7 Proof of Proposition 6

Recall that $\Delta = V^\dagger - C$. Both terms are monotone, but $V^\dagger$ is not submodular, so Δ inherits neither property. The instance of Proposition 13 witnesses both failures: $\Delta(\{s_1\}) = \Delta(\{s_2\}) = 0$ while $\Delta(\{s_1, s_2\}) = 1$, which violates diminishing returns, and adding a generalist to $\{s_1, s_2\}$ lowers Δ from 1 to $1 - 0.8 = 0.2$, which violates monotonicity.

For Eq. (7), fix $N \geq 2$, let $L = 2$, and take a pool of $N + 2$ members: generalists $g_1, \ldots, g_N$ with $B_{g_i} = (U_i, U_i)$ for independent $U_i \sim \text{Bernoulli}(1/2)$, and specialists with deterministic $B_{s_1} = (1,0)$ and $B_{s_2} = (0,1)$. Let $B_\Phi = (0,0)$, and let the pre-answer signal reveal $(U_1, \ldots, U_N)$.

At the pool level, $\bigvee_m B_m = \mathbf{1}$ always, because s_1 and s_2 cover both subgoals between them, while some member solves alone exactly when some $U_i = 1$. Hence $\Delta(\mathcal{M}) = \mathbb{P}[\bigwedge_i \{U_i = 0\}] = 2^{-N}$, which is at most ϵ once $N \geq \log_2(1/\epsilon)$.

At $k = 2$ we have $\mu_{g_i}(Z) = U_i$ and $\mu_{s_j}(Z) = \mu_\Phi(Z) = 0$. The best pair under F_Z is therefore two generalists, with $F_Z = \mathbb{E}[\max(U_1, U_2)] = 3/4$, and any pair containing a specialist is worse, while $V^\dagger(\{s_1, s_2\}) = 1$. Therefore $\max_{|S| \leq 2} V^\dagger(S) - \max_{|S| \leq 2} F_Z(S) = 1/4$, independently of N.

For the last claim, keep $N - 1 \geq 2$ generalists as above and replace the specialists by $B_{s_1} = (W, 0)$ and $B_{s_2} = (0, W)$, where $W \sim \text{Bernoulli}(1/2)$ is independent of the U_i and not revealed by Z. Collective solvability without individual solvability now requires $W = 1$ and all $U_i = 0$, so $\Delta(\mathcal{M}) = 2^{-N}$, exactly as in the first instance. At $k = 2$, however, $V^\dagger(\{s_1, s_2\}) = 1/2$, every

pair of one generalist and one specialist has $V^\dagger = 1/2$, and two generalists give $V^\dagger = F_Z = 3/4$. Hence $\max_{|S|\le 2} V^\dagger(S) = \max_{|S|\le 2} F_Z(S)$, and the assembly branch has no advantage. The two instances share the value of $\Delta(\mathcal{M})$ yet call for different branches under the budget, so no threshold rule on $\Delta(\mathcal{M})$ agrees with the budget-restricted comparison on both. □

C.8 Proof of Proposition 13 and Theorem 7

Increasing returns. In the instance of Theorem 4, the marginal gain of s_2 is 0 at $\emptyset$ and 1 at $\{s_1\}$, which violates diminishing returns. Monotonicity holds because enlarging S can only increase each factor $\max_{m\in S_\Phi} B_{m\ell}$.

Greedy failure. Set $U \sim \text{Bernoulli}(0.6)$ and keep everything else. In the first greedy step, $V^\dagger(\{g_i\}) = 0.6$ and $V^\dagger(\{s_i\}) = 0$, so greedy takes a generalist, say g_1. In the second step, $V^\dagger(\{g_1, g_2\}) = V^\dagger(\{g_1, s_1\}) = V^\dagger(\{g_1, s_2\}) = 0.6$, so greedy terminates at 0.6, while the optimum is $V^\dagger(\{s_1, s_2\}) = 1$. The ratio 0.6 is strictly below $1 - 1/e \approx 0.632$.

Inapproximability. Given a set cover instance with universe $[L]$, family $\{T_1, \dots, T_K\}$, and integer k, construct deterministic competence vectors $B_m = \mathbb{1}_{T_m}$, and let the anchor be correct on each subgoal independently with probability $\eta > 0$. If some S with $|S| \le k$ covers $[L]$, then $V^\dagger(S) = 1$. If none does, then every S with $|S| \le k$ leaves some subgoal ℓ uncovered, for which $\max_{m\in S_\Phi} B_{m\ell} = B_{\Phi\ell}$, so $V^\dagger(S) \le \mathbb{E}[B_{\Phi\ell}] = \eta$. Suppose a polynomial-time algorithm returns $\widehat{S}$ with $V^\dagger(\widehat{S}) \ge \alpha \max_{|S|\le k} V^\dagger(S)$ for a fixed $\alpha > 0$. Choosing $\eta < \alpha$, the returned value exceeds η if and only if a cover of size k exists, so the algorithm decides set cover, which is NP-complete (Feige, 1998). □

C.9 Proof of Lemma 14

Fix $(\mathcal{H}, D)$. By Assumption 8, the family $\{A_{m\ell}\}_{m,\ell}$ consists of independent Bernoulli variables. By Eq. (2), $B_{\pi(\ell)\ell} = \prod_{j\in \mathrm{an}^*(\ell)} A_{\pi(\ell)j}$ is a nondecreasing function of that family, so each event $\{B_{\pi(\ell)\ell} = 1\}$ is increasing. Harris's inequality for increasing events in independent Bernoulli variables gives, by induction on ℓ,

$$\mathbb{P}\Big[\bigcap_\ell \{B_{\pi(\ell)\ell} = 1\}\Big] \;\ge\; \prod_\ell \mathbb{P}\big[B_{\pi(\ell)\ell} = 1\big] \;=\; \prod_\ell \nu_{\pi(\ell)\ell},$$

where we suppress the conditioning. The argument is unchanged with A in place of B, since $A_{m\ell}$ is trivially increasing in itself. If G has no edges, then $\mathrm{an}^*(\ell) = \{\ell\}$, the events depend on disjoint sets of independent variables, and the inequality is an equality. □

Remark 21. Lemma 14 is why Theorem 9 does not require independence of the graded outcomes. Positive association can only increase the probability of a conjunction, so the independence form of the product is a lower bound. The corresponding claim for a selection composer would be false in this direction, which is another respect in which the branches differ.

C.10 Proof of Theorem 9

Fix ω and let $g_\ell(S)(\omega) = \max_{m\in S_\Phi} \nu_{m\ell}(\omega)$. By the argument of Theorem 1, each $g_\ell(\cdot)(\omega)$ is monotone submodular. By Assumption 8 and Eq. (2), $g_\ell \ge \nu_{\Phi\ell} = \prod_{j\in \mathrm{an}^*(\ell)} \alpha_{\Phi j} \ge \eta^{|\mathrm{an}^*(\ell)|}$, so g_ℓ takes values in $[\eta^{|\mathrm{an}^*(\ell)|}, 1]$. The function $\phi(t) = \log t$ is concave and nondecreasing there, and the composition of a concave nondecreasing function with a monotone submodular function is monotone submodular (Krause & Golovin, 2014, Sec. 1). Summing over ℓ and integrating over ω both preserve submodularity, so $\mathcal{L}$ is monotone submodular; subtracting $\mathcal{L}(\emptyset)$ normalizes it, and finiteness follows from $\eta > 0$.

For the lower bound, note that D is $\mathcal{H}$-measurable, so the assignment that picks a maximizer of $\nu_{\cdot\ell}$ at each subgoal is admissible in Eq. (4). Lemma 14 applied to this assignment gives $V_{\mathcal{H}}(S) \ge \mathbb{E}[\prod_\ell g_\ell(S)]$, and Jensen's inequality for the concave log gives $\log V_{\mathcal{H}}(S) \ge \mathbb{E}[\log \prod_\ell g_\ell(S)] = \mathcal{L}(S)$. Applying the same bound at $S = \emptyset$ gives $L_0 = \exp \mathcal{L}(\emptyset) \le V_{\mathcal{H}}(\emptyset)$.

Greedy on the normalized surrogate satisfies $\mathcal{L}(S^{\mathrm{gr}}) - \mathcal{L}(\emptyset) \geq (1 - 1/e)[\mathcal{L}(S^\star_{\mathcal{L}}) - \mathcal{L}(\emptyset)]$ by Nemhauser et al. (1978). Adding $\mathcal{L}(\emptyset)$, using $\log V_{\mathcal{H}}(S^{\mathrm{gr}}) \geq \mathcal{L}(S^{\mathrm{gr}})$, and exponentiating yields $V_{\mathcal{H}}(S^{\mathrm{gr}}) \geq L_0^{1/e}(\exp \mathcal{L}(S^\star_{\mathcal{L}}))^{1-1/e}$, which is Eq. (9) after dividing by L_0. □

C.11 Proof of Proposition 10

By Eq. (2), $B_{m\ell} \leq A_{m\ell}$ componentwise, so $F_B = \{\bigvee_m B_m = \mathbf{1}\} \subseteq F_A = \{\bigvee_m A_m = \mathbf{1}\}$. The integrands of the two expressions in Eq. (5) are $\mathbb{1}_{F_B}$ and $\mathbb{1}_{F_A}$, so their difference is $\mathbb{1}_{F_A \setminus F_B}$, which proves the first claim. Adding Eq. (6) gives Eq. (10). The two events are disjoint because $E_2 \setminus E_1 \subseteq F_B$, whereas $F_A \setminus F_B$ is disjoint from F_B. □

C.12 Proof of Proposition 17

Fix $(\mathcal{H}, D)$ and m. Taking expectations in Eq. (2) and using the independence of $\{A_{mj}\}_j$ from Assumption 8,

$$\nu_{m\ell} = \mathbb{P}\Big[\prod_{j \in \mathrm{an}^*(\ell)} A_{mj} = 1\Big] = \prod_{j \in \mathrm{an}^*(\ell)} \alpha_{mj},$$

so $\log \nu_{m\ell} = \sum_j \Gamma_{\ell j} \log \alpha_{mj}$, that is, $\log \nu_m = \Gamma \log \alpha_m$. In a topological order, $j \in \mathrm{an}(\ell)$ implies that j precedes ℓ, so $\Gamma_{\ell j} = 0$ for $j > \ell$ and $\Gamma_{\ell\ell} = 1$: the matrix is unit lower triangular. It is therefore invertible with $\det \Gamma = 1$ and integer inverse, the Möbius matrix of the reachability order, and Eq. (15) follows. For a chain, Γ is the lower triangular matrix of ones, whose inverse is the first-difference operator, giving $\log \alpha_{m\ell} = \log \nu_{m\ell} - \log \nu_{m,\ell-1}$ with the convention $\nu_{m0} = 1$. The prefix-support clause of Assumption 8 keeps the logarithms of all ancestor terms finite. □

C.13 Proof of Theorem 18

Submodularity follows from the argument of Theorem 9 with $\alpha_{m\ell}$ in place of $\nu_{m\ell}$: for fixed $(\mathcal{H}, D)$, the map $S \mapsto \max_{m \in S_\Phi} \alpha_{m\ell}$ is monotone submodular by Eq. (18), takes values in $[\eta, 1]$ by Assumption 8, and log is concave and nondecreasing there.

For the left inequality, Eq. (2) gives $\nu_{m\ell} \leq \alpha_{m\ell}$ pointwise, hence $\max_m \nu_{m\ell} \leq \max_m \alpha_{m\ell}$, and $\mathcal{L}(S) \leq \mathcal{L}^\ddagger(S)$ after taking logarithms, summing, and integrating. If at some subgoal ℓ no maximizer $m^\star$ of $\nu_{\cdot\ell}$ maximizes $\alpha_{\cdot\ell}$, then $\max_m \alpha_{m\ell} > \alpha_{m^\star\ell} \geq \nu_{m^\star\ell} = \max_m \nu_{m\ell}$; when this occurs with positive probability, the inequality is strict. Without edges $\nu = \alpha$ and the maximizers coincide, so strictness requires an ancestor.

For the right inequality, let π pick a maximizer of $\alpha_{\cdot\ell}$ at each subgoal; it is admissible because D is $\mathcal{H}$-measurable. By Lemma 14 applied to A,

$$V^\ddagger_{\mathcal{H}}(S) \;\geq\; \mathbb{E}\Big[\prod_\ell \alpha_{\pi(\ell)\ell}\Big] \;=\; \mathbb{E}\Big[\prod_\ell \max_{m \in S_\Phi} \alpha_{m\ell}\Big],$$

and Jensen's inequality for log gives $\log V^\ddagger_{\mathcal{H}}(S) \geq \mathcal{L}^\ddagger(S)$. The certificate then follows exactly as in the proof of Theorem 9, with $L_0^\ddagger = \exp \mathcal{L}^\ddagger(\emptyset)$. □

C.14 Proof of Proposition 19

Fix $(\mathcal{H}, D)$ and ℓ, and let $m^\star_\ell \in \arg\max_{m \in S_\Phi} \nu_{m\ell}$. By Proposition 17, $\nu_{m^\star_\ell \ell} = \alpha_{m^\star_\ell \ell} \prod_{j \in \mathrm{an}(\ell)} \alpha_{m^\star_\ell j}$. Since $\max_m \alpha_{m\ell} \geq \alpha_{m^\star_\ell \ell}$,

$$\log \max_m \alpha_{m\ell} - \log \max_m \nu_{m\ell} \;\geq\; \log \alpha_{m^\star_\ell \ell} - \log \nu_{m^\star_\ell \ell} \;=\; \sum_{j \in \mathrm{an}(\ell)} \log \frac{1}{\alpha_{m^\star_\ell j}} \;\geq\; 0,$$

where the last step holds because each $\alpha \leq 1$. Summing over ℓ and integrating gives Eq. (17). If G has no edges, every $\mathrm{an}(\ell)$ is empty and the bound is zero, consistent with $\nu = \alpha$. On a chain with all $\alpha_{mj} = \alpha$, we have $|\mathrm{an}(\ell)| = \ell - 1$, so the bound is $\log(1/\alpha) \sum_{\ell=1}^{L} (\ell - 1) = \binom{L}{2} \log(1/\alpha)$.

For the claim about specialists, Proposition 17 gives $\nu_{m\ell} = \alpha_{m\ell} \prod_{j \in \mathrm{an}(\ell)} \alpha_{mj}$, so the multiplicative understatement of member m at subgoal ℓ is exactly $\prod_{j \in \mathrm{an}(\ell)} \alpha_{mj}$, which decreases in the number

Algorithm 1 CAPS (composition-aware portfolio selection)

Input: pool $\mathcal{M}$, anchor Φ, DAG G, probe rate ρ, samples T, budget k or W, thresholds τ, τ_{prop}
Probe a random ρ fraction of cells with T draws each; issue prefix-forced probes on low-support cells
Deconvolve the observed rates by Eq. (15); fit Eq. (11) to the intrinsic targets and recover $\widehat{\nu}$
Fit cross-fitted heads $\widehat{\mu}_m^{(-f)}$ and $\widehat{\nu}_{m\ell}^{(-f)}$
Report $U_\Delta, U_{\text{prop}}$ from Propositions 2 and 10 ▷ build-or-not, not branch selection
for all available composer classes c, with $J_c \in \{\widehat{F}_Z^{\text{cf}}, \widehat{\mathcal{L}}, \widehat{\mathcal{L}^\ddagger}\}$ **do**
  $S_0^c \leftarrow \emptyset$; **for** $t = 1, \dots, k$: $S_t^c \leftarrow S_{t-1}^c \cup \{\arg\max_{m \notin S_{t-1}^c} [J_c(S_{t-1}^c \cup \{m\}) - J_c(S_{t-1}^c)]/w_m^{\mathbb{1}\{\text{knapsack}\}}\}$
  Execute composer c on the calibration split at each S_t^c; fit an attenuation curve $\widehat{a}_c(t)$
  $t_c^\star \leftarrow \arg\max_t \widehat{a}_c(t)\, \widetilde{J}_c(S_t^c)$, $R_c \leftarrow \widehat{a}_c(t_c^\star)\, \widetilde{J}_c(S_{t_c^\star}^c)$ ▷ $\widetilde{J}_c = J_c$, or e^{J_c} for log surrogates
end for
return $(c^\star, S_{t_{c^\star}^\star}^{c^\star})$ with $c^\star = \arg\max_c R_c$ ▷ exact over the operator family

of ancestors and in the member's weakness on them. In the instance of Theorem 4 extended to a chain, the specialist s_2, whose only intrinsic competence is at the terminal subgoal, has $\nu_{s_2 L} = 0$ while $\alpha_{s_2 L} = 1$. The uncorrected surrogate therefore assigns it no marginal value at any prefix, and greedy never selects it. □

C.15 PROOF OF PROPOSITION 11

Let ϵ be as in the statement, and let $\widehat{\pi} = \widehat{\pi}_S^{(-f)}$, which is measurable with respect to the out-of-fold data and therefore independent of the fold-f evaluation.

First, the argmax mismatch. Since $\widehat{\pi}(z)$ maximizes $\widehat{\mu}_\cdot^{(-f)}(z)$ over S_Φ, for the true maximizer $\pi^\star(z)$ we have

$$\mu_{\widehat{\pi}(z)}(z) \;\geq\; \widehat{\mu}_{\widehat{\pi}(z)}(z) - e(z) \;\geq\; \widehat{\mu}_{\pi^\star(z)}(z) - e(z) \;\geq\; \mu_{\pi^\star(z)}(z) - 2e(z),$$

where $\widehat{\mu} = \widehat{\mu}^{(-f)}$ and $e(z) = \max_{m \in S_\Phi} |\widehat{\mu}_m(z) - \mu_m(z)|$. Integrating and using $\mathbb{E}[e(Z)] \leq \epsilon$ gives $0 \leq F_Z(S) - \mathbb{E}[\mu_{\widehat{\pi}(Z)}(Z)] \leq 2\epsilon$. The bound does not involve $|S|$, because ϵ is defined over the whole pool and the chain holds pointwise in z.

Second, the evaluation error. Pointwise, $|\widehat{\mu}^{(f)}_{\widehat{\pi}(z)}(z) - \mu_{\widehat{\pi}(z)}(z)| \leq \max_m |\widehat{\mu}_m^{(f)}(z) - \mu_m(z)|$, so $|\mathbb{E}\widehat{F}_Z^{\text{cf}}(S) - \mathbb{E}[\mu_{\widehat{\pi}(Z)}(Z)]| \leq \epsilon$. By fold disjointness, the selected index is independent of $\widehat{\mu}^{(f)}$, so this term carries no selection-induced correlation; it vanishes when the fold-f head is conditionally unbiased, in which case the total bias is non-positive. Combining the two steps gives $|\mathbb{E}\widehat{F}_Z^{\text{cf}}(S) - F_Z(S)| \leq 3\epsilon$.

For the contrast, take $\mu_m \equiv \mu$ constant and $\widehat{\mu}_m = \mu + \sigma\xi_m$, with ξ_m independent standard Gaussians across the $|S| + 1$ members of S_Φ. Then $F_Z(S) = \mu$, while $\mathbb{E}[\max_{m \in S_\Phi} \widehat{\mu}_m] = \mu + \sigma\mathbb{E}[\max_{j \leq |S|+1} \xi_j] = \mu + \sigma(\sqrt{2\log|S|} + o(1))$, so the plug-in bias is positive and increasing in $|S|$. Because greedy compares sets of different cardinalities, this biases the comparison toward larger portfolios, the effect reported in Table 4.

Finally, if every marginal gain used by greedy is evaluated to within ϵ', the standard approximate-greedy analysis (Nemhauser et al., 1978; Krause & Golovin, 2014) gives $F_Z(S^{\text{gr}}) \geq (1 - 1/e)\max_{|S| \leq k} F_Z(S) - 2k\epsilon'$. Since each marginal gain is a difference of two estimates, taking $\epsilon' = 2(3\epsilon + \epsilon_n)$, with ϵ_n the Monte Carlo error of the n-point average, recovers a bound in terms of the nuisance error and the probe sample size. Constants are not optimized. □

D PSEUDOCODE

To facilitate reproduction, Algorithm 1 gives the pseudocode of CAPS.

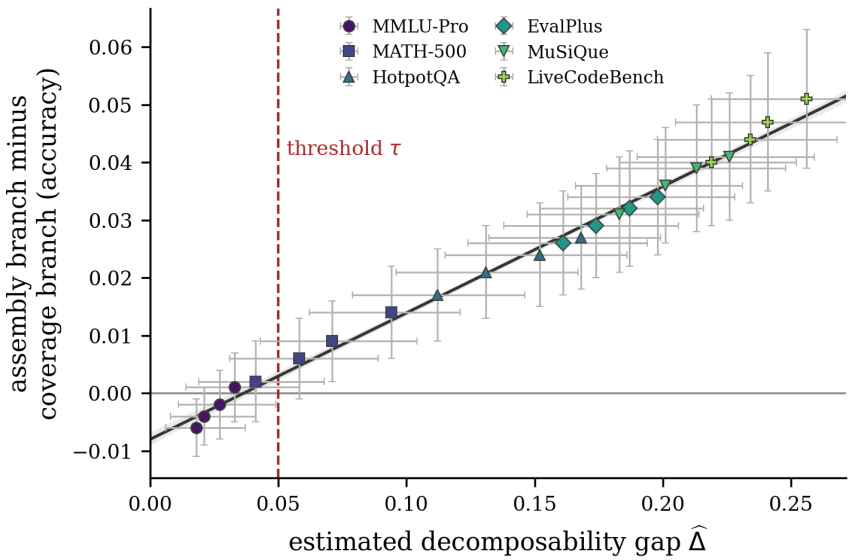


Figure 2: Diagnostic validation across benchmark–pool combinations. Cells to the left and right of the threshold correspond to the Self-MoA and mixed-portfolio regimes, respectively.

E Additional Experimental Results

E.1 Diagnostic validation

Figure 2 evaluates the diagnostic across 24 benchmark–pool combinations. Positive signed advantages favor the assembly portfolio, and negative values favor the coverage portfolio. The assembly advantage generally increases with the decomposability gap, as Eq. (6) suggests: a larger gap means more queries that the members solve collectively but none solves alone, and coverage cannot capture their value. The crossover near 0.05 marks an empirical change in which portfolio performs better across these configurations.

The three inconclusive cases have intervals that span this threshold, so their gap estimates do not place them clearly on either side of the crossover. They illustrate the uncertainty of threshold-based predictions near the boundary. Overall, the figure supports the gap as an empirical indicator of assembly gains in the evaluated pools. Under a portfolio budget, however, pool-level gaps can mislead (Proposition 6), so the final branch choice still relies on comparing completed portfolios on calibration data.

E.2 Component ablation

Table 4 removes one component at a time. Replacing the split-argmax estimator of Eq. (12) with the naive plug-in costs 2.1 points. More diagnostically, it raises the selected portfolio size by a median of 2, which is the signature of the size-dependent optimism that Proposition 11 removes. Removing low-rank completion and restricting greedy to fully probed candidates costs 1.7 points at equal probe budget. Removing multi-draw grading leaves accuracy almost unchanged but inflates the reported coverage ceiling by 4.6 points on average, so its effect is on the honesty of the evaluation rather than on the method itself. Removing the attenuation calibration costs 1.4 points, with the damage concentrated at large k, as the known non-monotonicity of real summarizers predicts. The propagation correction as a whole accounts for 1.9 points, of which prefix-forced probes contribute 0.8, so the deconvolution of Eq. (15) carries the larger share.

Reading the sign of $\Delta k^\star$. The last column separates two failure modes that the accuracy column conflates. Three variants overbuy. Without the split-argmax estimator, the plug-in maximum is optimistic by an amount that grows with portfolio size, so every marginal gain looks positive for longer than it should. Without the attenuation calibration, nothing reports the composer degrading under long contexts; the effect is larger, at $+3$, because the idealized objective is monotone by construc-

Table 4: Component ablation, averaged over six benchmarks and five selection seeds. Δ acc. is the change in realized accuracy relative to the full method and $\Delta k^\star$ the median change in the selected portfolio size. Every variant receives the same graded tensor and the same probe budget, so the differences are attributable to the component removed.

Variant	Realized acc.	Δ acc.	$\Delta k^\star$
CAPS (full)	.577 $\pm$.008	n/a	n/a
Naive plug-in maximum instead of Eq. (12)	.556 $\pm$.011	$-.021$	$+2$
No low-rank completion (probed candidates only)	.560 $\pm$.010	$-.017$	-1
No multi-draw grading ($T = 1$)	.574 $\pm$.013	$-.003$	0
No attenuation calibration	.563 $\pm$.009	$-.014$	$+3$
Assembly branch without the log surrogate	.551 $\pm$.012	$-.026$	$+1$
No propagation correction (surrogate fit on ν)	.558 $\pm$.010	$-.019$	-1
Deconvolution without prefix-forced probes	.569 $\pm$.009	$-.008$	0

tion. Without the log surrogate, greedy on $V^\dagger$ directly overbuys mildly, because its gains saturate late. Two variants instead underbuy. Removing low-rank completion restricts the candidate set to fully probed members, which mechanically shortens the greedy path. Removing the propagation correction underbuys for the reason Proposition 19 gives: members whose value lies at deep subgoals are scored through their own upstream failures, their marginal gains vanish, and greedy stops early. The two modes are therefore distinguishable before any accuracy is measured. This is useful in deployment, where the realized accuracy of a candidate portfolio is expensive to obtain but its selected size is free.

E.3 GENERALIZATION AND STABILITY

Table 5: Generalization and stability. *Held-out models* re-runs selection on a family-disjoint subset of the pool, so that the two parts share no model family; *held-out distribution* transfers a portfolio to a task mixture shifted toward atomic items without re-selection. Jaccard is the mean pairwise overlap of selected portfolios across five folds.

Method	In distribution	Held-out models	Held-out distribution	Jaccard
Top-k accuracy	.528 $\pm$.009	.519 $\pm$.012	.491 $\pm$.015	.61
Min error corr.	.530 $\pm$.011	.514 $\pm$.014	.478 $\pm$.018	.34
Greedy coverage	.538 $\pm$.008	.527 $\pm$.011	.495 $\pm$.014	.52
Model-first greedy	.548 $\pm$.009	.536 $\pm$.012	.508 $\pm$.015	.58
CAPS (ours)	**.577** $\pm$.008	**.568** $\pm$.010	**.546** $\pm$.013	**.79**

Table 5 tests two splits that the routing literature does not usually separate. Holding out a family-disjoint part of the pool and re-running selection changes the realized accuracy of CAPS by 0.9 points and that of the baselines by 0.9 to 1.6 points, so none of the methods relies on a particular set of checkpoints. We split by family rather than at random for the reason given in Appendix F: a random split leaves near-duplicate siblings on both sides and would overstate transfer, since a method could recover most of its portfolio by substituting a different size of the same lineage.

Holding out a task distribution is considerably harder. A portfolio chosen on the four benchmarks with $\widehat{\Delta} > 0.15$ and evaluated on a mixture shifted toward atomic items loses 3.1 points relative to re-selection in the new mixture, although it still outperforms every baseline under the same shift. This failure mode is expected. Both gaps in Eq. (10) are properties of the task distribution rather than of the pool, so a shift in the mixture moves the target against which the portfolio was optimized. The observation also motivates the distributionally robust extension discussed in Appendix H.

Stability across folds is 0.79 for CAPS, against 0.52 for greedy coverage and 0.34 for minimum error correlation. We attribute the gap to the surrogate of Eq. (8) being smoother in the estimated competences than a hard coverage count. Coverage changes only when a member crosses the correctness threshold on some item, so small perturbations of the fold either leave the objective untouched or flip it discontinuously; the logarithmic surrogate instead moves continuously. Stability matters in-

dependently of mean accuracy. A rule that returns a different portfolio on every fold cannot be deployed, because the retained models must be warmed, contracted for, and kept resident, and these commitments cannot be revised per fold.

F Model Pool

Table 6 lists the 42 checkpoints of the live pool. Selection criteria were fixed before any grading: nine model families, a parameter range spanning more than two orders of magnitude, and at least two checkpoints per family, of different sizes wherever the family provides them, so that capability and family are not confounded. We include reasoning-tuned and domain-tuned members of the same lineage, which are the natural source of the intrinsic subgoal specialization that Theorem 4 concerns.

The cost weight w_m used by the knapsack variant is the measured median per-item wall time, normalized by that of Qwen2.5-14B-Instruct, a mid-sized reference checkpoint. Latency is reported separately for the two inference paths, since they differ by more than an order of magnitude and a single scalar per model would misstate the cost-aware analysis. The scoring path is one forward pass over answer-label logits for multiple-choice items, and the generation path decodes up to the per-benchmark token cap.

Two properties of the pool matter for reading the results. First, the best fixed model differs across benchmarks, and the largest checkpoint is not the best on all six, so the accuracy ordering is not a proxy for parameter count. Second, latency is not monotone in parameter count. On the generation path, the mixture-of-experts members and the reasoning-distilled members invert the ordering: the former activate only a fraction of their parameters, and the latter emit far more tokens. This is why the cost axis in Figure 1a is measured rather than inferred from size.

For the held-out-model experiment of Table 5, the pool is split by family rather than at random. The retained families (25 checkpoints) are Qwen2.5, Gemma, Phi, InternLM2.5, and OLMo-2, and the held-out families (17 checkpoints) are Llama-3, Mistral, DeepSeek, and GLM-4. The only cross-half link is that the R1-Distill-Qwen checkpoints are distilled onto Qwen2.5 bases; we group them under DeepSeek by their distillation source. Splitting at random would leave near-duplicate siblings on both sides and would overstate transfer.

G Implementation Details

G.1 Generation, grading, and stochasticity

All members use their documented chat template with no few-shot exemplars. Generation uses nucleus sampling at temperature 0.7 with $T = 5$ independent draws per cell, under the per-benchmark token cap in Table 7. The multiple draws explain why our ceilings are lower than single-draw ceilings. With one draw, a member that is correct with probability one half contributes a full unit of coverage half of the time and nothing otherwise, so the maximum over members inherits an upward bias that grows with portfolio size. We therefore target the per-cell success probability and examine single-draw grading only as an ablation (Table 4). Parse failures count as incorrect. This conservative choice charges a member for an unreadable answer rather than silently shrinking its denominator. Multiple-choice items are also scored without generation, by one forward pass over the first tokens of the option labels, and the two scoring modes are reported as a robustness check.

Every artifact stores, per member and item, the raw prompt, the tokenized prompt, the raw output, the parsed answer, the gold answer, the per-subgoal grade vector, the token counts, and the parser status. A single script emits all reported numbers from these stored matrices. This matters because rounding the stored confidences to four decimals is by itself enough to change argmax ties and shift a confidence-gated baseline by more than a point.

G.2 Subgoal extraction and prerequisite DAGs

Prerequisite structure is read from benchmark metadata rather than learned, so that a misspecified graph cannot be blamed on our estimator. Table 7 records, for each benchmark, the subgoal unit, the grader, the DAG shape, and the resulting mean subgoal count and mean prerequisite depth.

Table 6: Live candidate pool: 42 instruction-tuned checkpoints from nine families, all evaluated in bfloat16 on A100 GPUs with the documented chat template, using tensor parallelism for checkpoints that exceed one device. Revisions are truncated commit hashes, resolved against the released revision file. Latency is the per-item median for the scoring and generation paths. w_m is the normalized serving cost used under a knapsack budget.

Checkpoint	Family	Params	Revision	Latency (score / gen)	w_m
Qwen2.5-0.5B-Instruct	Qwen2.5	0.5B	`7ae5576`	21 ms / 4.1 s	0.18
Qwen2.5-1.5B-Instruct	Qwen2.5	1.5B	`989aa79`	25 ms / 5.6 s	0.29
Qwen2.5-3B-Instruct	Qwen2.5	3B	`aa8e725`	34 ms / 7.2 s	0.41
Qwen2.5-7B-Instruct	Qwen2.5	7B	`bb46c15`	47 ms / 9.4 s	0.62
Qwen2.5-14B-Instruct	Qwen2.5	14B	`cf98814`	78 ms / 14.8 s	1.00
Qwen2.5-32B-Instruct	Qwen2.5	32B	`5ede1c9`	141 ms / 26.1 s	1.84
Qwen2.5-72B-Instruct	Qwen2.5	72B	`d3d9510`	288 ms / 51.7 s	3.71
Qwen2.5-Coder-7B-Instruct	Qwen2.5	7B	`c1a1a4e`	47 ms / 9.1 s	0.61
Qwen2.5-Coder-32B-Instruct	Qwen2.5	32B	`7e0b2f3`	140 ms / 25.8 s	1.82
Qwen2.5-Math-7B-Instruct	Qwen2.5	7B	`4a83ca6`	46 ms / 9.9 s	0.63
Qwen2.5-Math-72B-Instruct	Qwen2.5	72B	`f0a2c88`	287 ms / 53.0 s	3.76
QwQ-32B	Qwen2.5	32B	`976039c`	142 ms / 41.3 s	2.55
Llama-3.2-1B-Instruct	Llama-3	1B	`9213176`	23 ms / 4.6 s	0.22
Llama-3.2-3B-Instruct	Llama-3	3B	`0cb88a4`	32 ms / 6.4 s	0.38
Llama-3.1-8B-Instruct	Llama-3	8B	`0e9e39f`	51 ms / 10.2 s	0.67
Llama-3.1-70B-Instruct	Llama-3	70B	`945c890`	279 ms / 49.8 s	3.58
Llama-3.3-70B-Instruct	Llama-3	70B	`6f6073b`	281 ms / 50.4 s	3.62
gemma-2-2b-it	Gemma	2B	`299a856`	49 ms / 7.8 s	0.44
gemma-2-9b-it	Gemma	9B	`8c8c5e4`	88 ms / 13.1 s	0.79
gemma-2-27b-it	Gemma	27B	`2f3b3e1`	166 ms / 24.6 s	1.71
codegemma-7b-it	Gemma	7B	`ba5a4f0`	74 ms / 11.2 s	0.68
Mistral-7B-Instruct-v0.3	Mistral	7B	`c170c70`	49 ms / 9.0 s	0.60
Mistral-Nemo-Instruct-2407	Mistral	12B	`4b5a6e1`	69 ms / 12.4 s	0.86
Mistral-Small-Instruct-2409	Mistral	22B	`4a0a5c8`	112 ms / 19.9 s	1.39
Mixtral-8x7B-Instruct-v0.1	Mistral	47B	`41bd4c9`	96 ms / 17.3 s	1.61
Mixtral-8x22B-Instruct-v0.1	Mistral	141B	`b1b2c3d`	214 ms / 38.2 s	3.94
Phi-3-mini-4k-instruct	Phi	3.8B	`2fe1924`	46 ms / 8.5 s	0.52
Phi-3-small-8k-instruct	Phi	7B	`9f2a7be`	62 ms / 11.0 s	0.71
Phi-3.5-MoE-instruct	Phi	42B	`482a12f`	91 ms / 16.4 s	1.52
phi-4	Phi	14B	`187ef06`	80 ms / 15.2 s	1.03
DeepSeek-Coder-V2-Lite-Inst.	DeepSeek	16B	`e434a23`	58 ms / 10.6 s	0.74
R1-Distill-Qwen-1.5B	DeepSeek	1.5B	`ad9f0ae`	26 ms / 8.8 s	0.37
R1-Distill-Qwen-7B	DeepSeek	7B	`916b56a`	48 ms / 22.6 s	1.12
R1-Distill-Qwen-32B	DeepSeek	32B	`3865e12`	143 ms / 44.1 s	2.68
R1-Distill-Llama-70B	DeepSeek	70B	`6ede5f5`	282 ms / 78.4 s	4.91
internlm2_5-1_8b-chat	InternLM2.5	1.8B	`b4b4e1c`	27 ms / 5.4 s	0.27
internlm2_5-7b-chat	InternLM2.5	7B	`3e1f0a4`	50 ms / 9.6 s	0.64
internlm2_5-20b-chat	InternLM2.5	20B	`ac1d4b2`	108 ms / 19.1 s	1.33
glm-4-9b-chat	GLM-4	9B	`cb0da49`	86 ms / 13.4 s	0.81
codegeex4-all-9b	GLM-4	9B	`80f31e6`	85 ms / 13.0 s	0.79
OLMo-2-1124-7B-Instruct	OLMo-2	7B	`c1fa1c1`	52 ms / 10.0 s	0.66
OLMo-2-1124-13B-Instruct	OLMo-2	13B	`e3d0b09`	77 ms / 14.3 s	0.98

Table 7: Subgoal units, graders, and prerequisite structure per benchmark. Depth is the number of edges on the longest prerequisite path of an item, averaged over items. The two code suites have depth one because the tests share a single program but do not depend on one another.

Benchmark	Subgoal unit	Grader	DAG shape	$\bar{L}$ / $\bar{d}$	Token cap
MMLU-Pro	whole item	exact match	singleton	1.0 / 0.0	512
MATH-500	proof step	process RM	chain	6.8 / 5.8	2048
HotpotQA	supporting fact	span EM	shallow tree	2.4 / 1.4	512
MuSiQue	sub-question	span EM	composition	3.1 / 2.0	512
EvalPlus	unit test	test harness	depth-one star	14.2 / 1.0	1024
LiveCodeBench	unit test	test harness	depth-one star	21.6 / 1.0	2048

For the two code suites, the closed reachability matrix is the identity plus one column for a shared root that represents successful compilation. The inversion of Eq. (15) then reduces to dividing each per-test rate by the compilation rate. For MATH-500, the graph is the linear order of graded steps, and the inversion is the prefix-conditional ratio of Proposition 17. For the two multi-hop sets, the

Table 8: Hyperparameters. Everything in this table was fixed on the probe split before test outcomes were opened.

Component	Setting	Value	Selected by
Probing and completion	rank r of Eq. (11)	16	probe-split likelihood
	ℓ_2 penalty	10^{-3}	probe-split likelihood
	optimizer	Adam, lr 10^{-3}, 200 epochs	fixed
	probe rate ρ	0.04 live pool, 0.008 RouterEval	Fig. 1b
	prefix-support threshold ζ	0.05	fixed a priori
Heads, Stage 2	encoder	`gte-large`, frozen	fixed
	robustness encoder	`all-mpnet-base-v2`	fixed
	architecture	2-layer MLP, K outputs, width 512	probe split
	cross-fitting	5 folds, shared with Eq. (12)	fixed
Diagnostic, Stage 3	interval level δ	0.05, Clopper and Pearson	fixed
	τ	0.05	fixed a priori
	τ_{prop}	0.05	fixed a priori
Greedy, Stage 4	evaluation	lazy, cost-benefit under knapsack	fixed
	cost weights w_m	Table 6	measured
	size sweep	$k \in \{1, \ldots, 8\}$, attenuation-calibrated	Fig. 1a
Reporting	draws per cell T	5 at temperature 0.7	fixed
	selection seeds	5	fixed

graph is the annotated composition order, which is a tree for HotpotQA and a general DAG for MuSiQue.

Prefix-forced probes are issued only where the estimated prefix support falls below $\zeta = 0.05$, which in our runs covers 7.4 percent of cells. In code, the injection replaces every upstream module with a reference implementation and re-runs only the tests downstream of it. In mathematics, it supplies the graded reference solution up to the parent step and asks for a continuation. In multi-hop question answering, it supplies the gold answers to the parent sub-questions. The mathematics and multi-hop probe prompts appear in Appendix G.5. No prefix-forced probe is issued on a test item.

G.3 Estimators, optimizer, and thresholds

Table 8 collects the settings, two of which deserve comment. First, the rank of the completion model is selected on the probe split alone and never on anything that touches the ladder in Table 1. The diagnostic thresholds these ceilings, so tuning a nuisance model against them would invalidate the thresholds. Second, head fitting and the split-argmax estimator of Eq. (12) share the same five folds. This sharing makes the selected index independent of the evaluated value in Proposition 11; independent fold assignments for the two purposes would reintroduce the correlation that the estimator exists to remove.

Diagnostic interpretation and screening. Upper confidence limits make the diagnostic conservative about dismissing potentially useful composer capabilities. Specifically, $U_\Delta \leq \tau$ indicates limited distributed-solvability headroom in the probed pool, whereas $U_{\text{prop}} > \tau_{\text{prop}}$ flags propagation repair for further evaluation. The latter does not establish a realized gain, because Δ_{prop} is a difference between ideal ceilings (Proposition 10). The confidence level and thresholds are fixed as specified in Table 8. When the composer calls required for calibration are unaffordable, the diagnostic can serve as a screening shortcut. This use is heuristic, since a pool-level threshold need not agree with the budget-restricted branch comparison (Proposition 6). Section 6, and Table 3 in particular, evaluates the shortcut on the original and augmented pools and quantifies its sensitivity to dilution.

G.4 Composer configurations

We instantiate three composer classes, one per branch of Algorithm 1. All are built on the same anchor, so that differences are attributable to the composition rule rather than to anchor strength.

The **selection composer** sees only the pre-answer signal, invokes exactly one member, and returns that member's answer verbatim. The **splice-free assembly composer** invokes every member of the

portfolio, receives the answers and rationales in an order randomized per item to remove position effects, and writes a new answer. The **splice-capable assembly composer** additionally receives the per-subgoal decomposition, selects a source member for each subgoal, and re-executes the downstream steps on the repaired prefix. In code, this means re-running the test suite against a program assembled from the selected modules; in mathematics, it means continuing from the selected partial solution. The three composers sit in different cost regimes, so Figure 1a reports measured cost rather than invocation count.

G.5 PROMPT TEMPLATES

Templates are shown with placeholders in angle brackets. Each was fixed before grading and applied identically to every member.

Proposer, multiple choice

```
Answer the question.  Think briefly, then end your reply with the line
"ANSWER: X" where X is the letter of one of the options.
Question:  <question> A. <option a> B. <option b> ...  J. <option j>
```

Proposer, mathematics with step markers

```
Solve the problem.  Number each step as "Step k:" on its own line,
and end with the line "ANSWER: <value>".  Do not skip steps and do not
merge two steps into one line.
Problem:  <problem>
```

Proposer, code

```
Complete the specification below.  Return only executable code, with
no Markdown fences and no commentary.  Define every helper you use.
Specification:  <signature and docstring>
```

Splice-free assembly composer

```
You are given <k> candidate answers to the same question, each with
its reasoning.  They may disagree and some may be wrong.  Write the
best answer you can.  You may combine partial reasoning from several
candidates, and you may reach a conclusion that none of them reached.
End with the line "ANSWER: <value>".
Question:  <question>
Candidate 1:  <answer and rationale> Candidate 2:  <answer and
rationale> ...  Candidate <k>:  <answer and rationale>
```

Splice-capable assembly composer

```
You are given <k> candidate solutions to the same problem, decomposed
into the subgoals listed below.  For each subgoal, choose the
candidate whose work on that subgoal you judge correct, then
rewrite the solution using the chosen pieces, re-deriving everything
downstream of any piece you replaced.
Report your choices as lines "SUBGOAL k:  candidate j" before the
rewritten solution, and end with "ANSWER: <value>".
Problem:  <problem> Subgoals:  <subgoal 1>, <subgoal 2>, ..., <subgoal
L>
Candidate 1:  <solution, segmented by subgoal> ...  Candidate <k>:
<solution, segmented by subgoal>
```

Prefix-forced probe, mathematics

```
Below is the beginning of a correct solution, up to and including Step
<j>.  Continue from Step <j+1> to the end.  Do not modify or restate
the steps you were given.  End with the line "ANSWER: <value>".
```

```
Problem: <problem> Correct prefix: <reference steps 1 through j>
```

Prefix-forced probe, multi-hop question answering

```
Answer the final question. The answers to its prerequisite
sub-questions are given and are correct; use them as stated and do
not re-derive them.
Prerequisites: <sub-question: gold answer> ... Final question:
<question>
```

LLM-judge baseline

```
Rate how likely each candidate answer is to be correct, on an integer
scale from 1 to 5. Output one line per candidate as "CANDIDATE j:
s". Do not explain.
Question: <question> Candidate 1: <answer and rationale> ...
Candidate <k>: <answer and rationale>
```

G.6 Baselines and compute

Every baseline receives the same completed tensor, encoder, composer, and portfolio size, so differences are attributable to the selection objective alone. Approximate Shapley values use 40 sampled coalitions of sizes one through four with 20 items per coalition, the largest budget under which the method remains cheaper than our full pipeline. The diversity baselines operate on end-to-end error vectors rather than subgoal vectors, because that is how the literature defines them; running them on subgoal vectors is reported as a variant and does not change the ordering. Greedy complementary coverage is anchored at the same Φ as ours, so it is a faithful instantiation of Shihab et al. (2026) rather than a weakened version.

Total compute is dominated by Stage 1. Grading the live pool at $\rho = 0.04$ with $T = 5$ draws costs roughly 1.9 thousand A100 hours, and prefix-forced probes add 6 percent to that. The completion model, the heads, and all greedy runs together take under 20 A100 hours, and the composer calibration calls take under 40. This asymmetry is the practical argument for the pipeline: the expensive object is the graded tensor, which is shared across branches and across all baselines, and everything the method adds on top of it is cheap.

H Extended Limitations and Future Work

The conjunctive model. Eq. (1) is exact for test-suite grading, where passing is by definition the conjunction of per-test outcomes, and reasonable for annotated multi-hop decompositions. For open-ended generation, the subgoals would be rubric items whose conjunction is a modeling convention rather than the definition of correctness, and our framework does not say which conjunction is right. We therefore restrict the live experiments to benchmarks whose grader is itself conjunctive, and we regard extension to rubric-graded generation as an open problem rather than a straightforward instantiation.

Grader dependence. Subgoal labels are cheap for code, moderately expensive and noisy for step-graded mathematics, and unavailable at scale for anything that requires human decomposition. Both gaps in Eq. (10) inherit any bias the grader carries. For example, a process reward model that systematically credits plausible but wrong steps inflates the intrinsic competence of members that produce such steps. Our multi-draw protocol controls sampling noise but not grader bias.

Propagation assumptions. The correction introduces three new requirements. First, the prerequisite DAG is taken as given; this is defensible for our benchmarks, but a misspecified reachability matrix propagates directly into Eq. (15). Second, Assumption 8 requires local independence of intrinsic competences given latent difficulty, the standard item-response condition. It fails when two members share a systematic misconception at one subgoal, and in that regime the deconvolution overstates intrinsic complementarity. Third, the prefix-support clause is the binding practical constraint: prefix-forced probing is cheap for code and tractable for step-graded mathematics but has no obvious analogue for open-ended generation.

Gaps in the guarantees. Remark 15 shows that the splice-free branch discards a nonnegative propagation surplus in exchange for submodularity; Lemma 14 certifies the direction of this loss, but we do not bound its magnitude. The assembly guarantee is multiplicative in the log rather than a constant factor. By Theorem 7 no better guarantee is available in general, but instance-dependent bounds through a submodularity ratio of $V^{\dagger}$ would be sharper than what we prove. Finally, the joint search over portfolio and operator is exact only relative to the operator family we instantiate. Three classes with one fixed composer each is a coarse discretization of a space that also includes the prompt, the layer count, the verification strategy, and the debate protocol.

Static portfolios. We select once. New checkpoints arrive, old ones are withdrawn or repriced, and switching a served model carries real cost, so a deployment that rebalances monthly falls outside our formulation. Treating the portfolio as an online decision with switching cost turns Theorem 1 into a question about regret under a submodular reward, and the model-level factors of Eq. (11) already supply a cold-start representation for newly released candidates.

Other extensions. Portfolios chosen on one task mixture transfer imperfectly, as Table 5 shows. Replacing the expectation in Eq. (3) with a conditional value at risk over an ambiguity set of query distributions would target the worst-case mixture directly. Learning the prerequisite structure from trajectories, within the identifiability limits that the prefix-support clause already hints at, would extend the correction to settings without subgoal annotations. Finally, choosing which cells to probe is a combinatorial pure-exploration problem with a sum-max reward, where sample-complexity bounds would replace our uniform probing with an adaptive scheme.